MASTER EN SCIENCES INFORMATIQUES

Étude comparative d'architectures de réseau par rapport au modèle de référence OSI de l'ISO

Meyer, Jean-François

Award date:
1982

Awarding institution:
Universite de Namur

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR,

INSTITUT D'INFORMATIQUE

ETUDE COMPARATIVE D' ARCHITECTURES DE RESEAU PAR RAPPORT AU MODELE DE REFERENCE OSI DE L' ISO

Promoteur : Ph. Van Bastelaer.

Meyer Jean-François

Mémoire présenté en vue
de l'obtention du grade de

LICENCIE ET MAITRE EN INFORMATIQUE

ANNEE ACADEMIQUE 1981 - 1982.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR,

INSTITUT D'INFORMATIQUE

ETUDE COMPARATIVE D'ARCHITECTURES DE RESEAU PAR RAPPORT AU MODELE DE REFERENCE OSI DE L'ISO

Promoteur : Ph. Van Bastelaer.

Meyer Jean-François

Mémoire présenté en vue
de l'obtention du grade de

LICENCIE ET MAITRE EN INFORMATIQUE

ANNEE ACADEMIQUE 1981 - 1982.

E R R A T A

page	ligne	lire	au lieu de
1. Bibliographie			
4	ref 40	transfer	transfeer
2. CHAPTER 1 :			
1 - 3	5	communicate	communicates
3. CHAPTER 2 :			
2 - 1	12	buys	buy
2 - 1	32	medias	media
2 - 3	33	standards	standard
2 - 3	38	OSI RM	OSI 80
2 - 5	16	in the	in
2 - 6	last line	running	runing
2 - 8	16	useful	usefull
2 - 10	38	contain	contains
2 - 12	11	adapted from [Tanenbaum 81].	
2 - 13	18	own	owns
2 - 15	last line	are not architectured and are product dependents.	
2 - 16	1	plays	play
2 - 17	17	example	exemple
2 - 22	32	of	off
2 - 23	9	vendors	the vendors
2 - 24	12	consistent	consistents
2 - 24	14	service	services
2 - 24	20	provide	provides
2 - 24	6	semantics	sementics
2 - 25	18	three	two
2 - 25	30	networks seems	network seem
2 - 25	35	addresses	address
2 - 30	5, 6	network	net
2 - 31	5	advantage	advantages
2 - 38	10	unprobable	unprobalble

4. CHAPTER 3 :

3 - 1	24	reasons	reason
3 - 9	14	send	sends
3 - 10	28	therefore	therefor
3 - 15	7	except	exepts
3 - 15	25	between	betwenn
3 - 18	32	model	mold
3 - 20	21	topology	toology
3 - 35	17	Throughout	Throught

5. Table des matieres de la partie pratique

5.4.1. Mapping, "initiator address" et "acceptor address", transport address

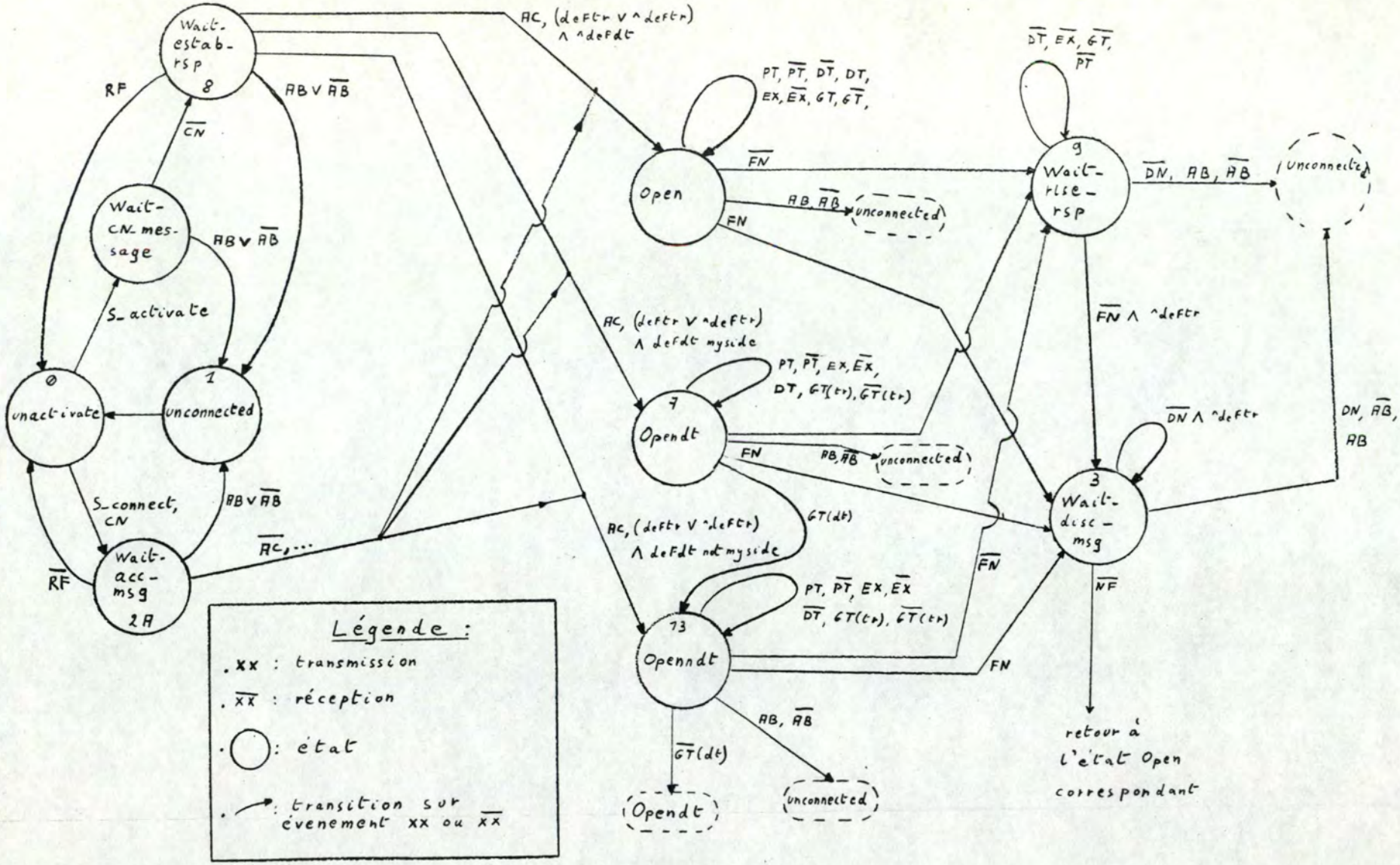
au lieu de

5.4.1. Mapping, transport address

6. SECONDE PARTIE

page	ligne	lire	au lieu de
2	7	abrèviation	abrèviations
4	29	B 2	L 2
19	2	connexion-session	connexion-transport

5.5.4. Diagramme d'états & transitions (fig. 9).



6.6.3. Diagramme d'états et transitions.

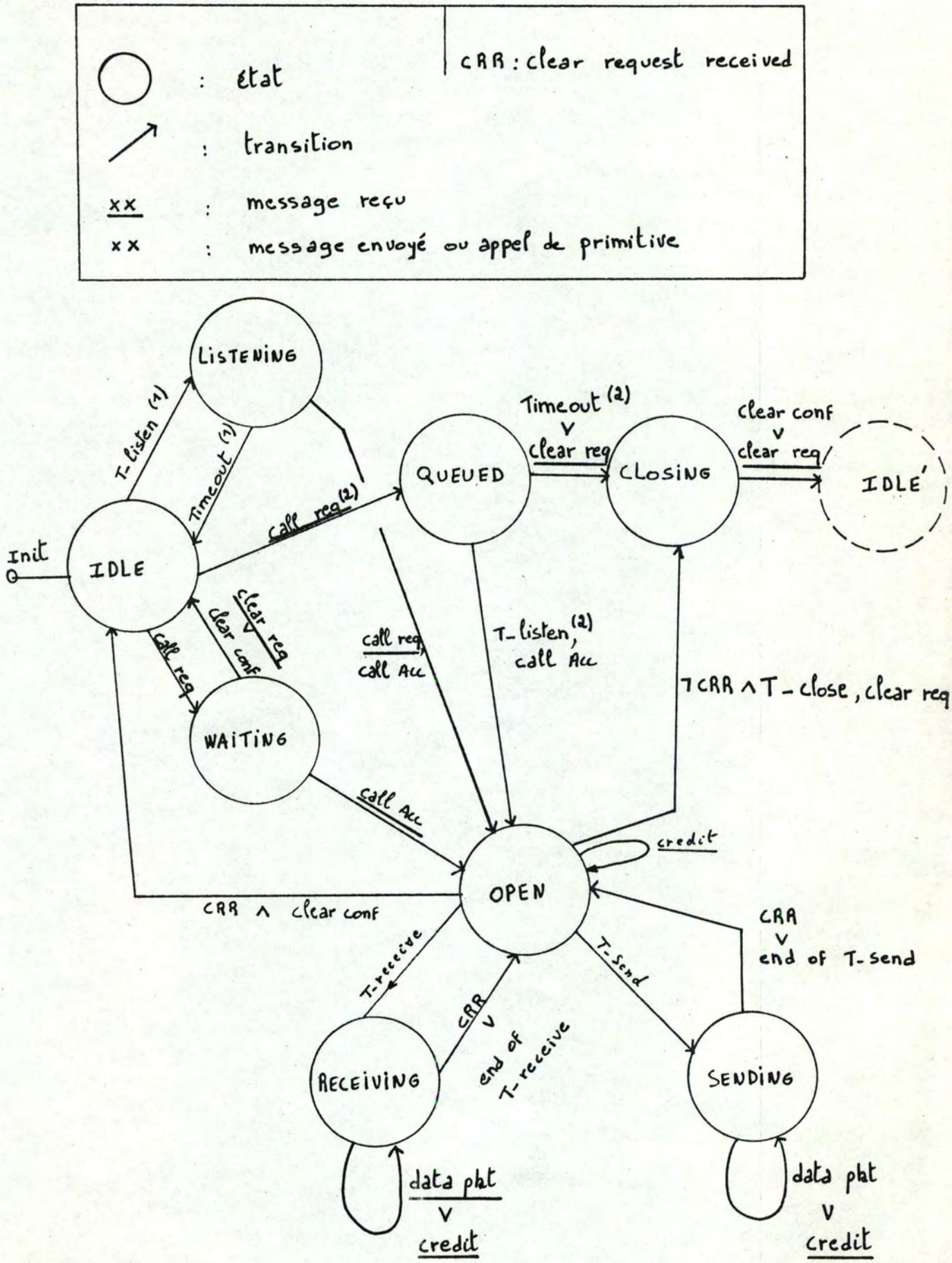


Figure 11 : diagramme d'états et transitions

-- REMERCIEMENTS --

Je tiens à remercier, pour les conseils et l'aide qu'ils m'ont apportés dans la réalisation de ce mémoire:

Monsieur Ph. Van Bastelaer : promoteur du mémoire

Messieurs J-L Bonato,
M. Deuwel,
Ph. Gransart : qui m'ont pris en charge durant le stage.

Monsieur Verhaeghe René : lors de la réalisation pratique de la maquette.

Je remercie également mes parents qui m'ont permis de réaliser ces études.

-- A V A N T - P R O P O S --

Ce mémoire se compose de deux parties distinctes:

La première, rédigée en anglais, constitue une approche du modèle de référence OSI de l'ISO ainsi qu'un début d'étude comparative de quelques architectures de réseau en relation avec ce modèle.

La seconde, rédigée en français, est la description d' une maquette visant à simuler les 3 couches supérieures du modèle OSI, et plus particulièrement les niveaux Transport et Session.

Les tables des matières de ces deux parties y sont attenantes. La bibliographie est placée à la suite de cet avant-propos afin de faciliter les consultations.

-- B I B L I O G R A P H I E --

1. AHUJA 79
   Ahuja V.
   Routing and flow control in Systems Network Architectures
   IBM Syst. Journ. vol 18 No 2
   p 298 - 314
   1979

2. CII HB
   DSA
   CII HB

3. CNA DDR
   CNA Reference Model
   NCR Standard n. 7-10-01
   March 82

4. CORR NEAL
   Corr F.P. & Neal D.H.
   SNA and emerging international standards
   IBM Syst. Journ. vol 18 No 2
   p 244 - 259
   1978

5. CORR NEAL
   Corr F. P. & Neal D. H.
   SNA and Emerging International Standards
   IBM System Journal - vol 18 - No 2 - p. 244-259
   1979

6. CYPSER 78
   Cypser R. J. IBM
   Communication Architecture for Distributed Systems
   Addison-Weysley Publishing Company
   1978

7. DANTH 78
   Danthine A. & Bremer J.
   Modeling and Verification of End-to-End Transport Protocols
   Computer Network Protocols - Danthine Editor - p. F5 1-12
   1978

8. DC IBM
   Taylor F. E.
   Data Communications with IBM
   Systems Technology Consultants

9. DCBC
Data communications - Basic concepts
Datapro R.C. C05-010-101
p 101 - 108
January 80

10. DEC DAP
DNA Data Access Protocol Functional Specifcations
DEC AA-K177A-TK
October 80

11. DEC GD
DNA General Description
DEC AA-K179A-TK
October 80

12. DEC INT
Introduction to Decnet DEC No AA-J055-TK
January 80

13. DEC MOP
DNA Maintenance Operation Protocol Functional Specifications
DEC AA-K178A-TK

14. DEC NM
DNA Network Management Functional Specifications
DEC AA-K181A-TK
October 80

15. DEC NSP
DNA Network services Functional Specifications
DEC AA-K176A-TK
October 80

16. DEC RSX
RSX Decnet user´s guide
DEC No AA-H223A-TC, vol 1
January 80

17. DEC SC
DNA Session Control Functional Specifications
DEC AA-K182A-TK
October 80

18. DEC TFS
DNA Transport Functional Specifications
DEC AA-K180A-TK
October 80

19. DIDCS
Direction for international data communication standards
Datapro Research Corporation CS93-105-108
1981

20. DNA DTP
Dec Digital Network Architecture
Datapro R.C., C11-384-101
p 101-114
May 80

21. DP DSA
Honeywell DSE and DSA
Datapro Research Corporation - C11-480-101 - p. 101-115
Augustus 81

22. DP NAPDP
Green
Network Architecture and Protocols for Distributed Processing
Datapro Research Corporation - CS20-508-101 - p. 101-112
July 81

23. DP SNA
IBM System Network Architecture
Datapro Research Corporation - C11-491-101 - p. 101-110
December 80

24. DSA GD
DSA General Description
CII-HB

25. ECMA 40
HDLC Frame Structure
ECMA 40
January 80

26. ECMA 49
HDLC Element of Procedure
ECMA 49
Augustus 79

27. ECMA 60
HDLC Unbalaced Class of Procedure
ECMA 60
Augustus 79

28. ECMA 61
HDLC Balanced Class of Procedure
ECMA 61
Augustus 79

29. ECMA 72
Transport Protocol
ECMA 72
January 81

30. ECMA 98
Bourguignon
Proposed Tools for the Definition of the Session Protocol
ECMA/TC23/80/98 - p. 1-11
June 80

31. ECMA DPP
Data Presentation Protocol (first draft)
ECMA/TC23/81/141 - p. 1-11
1981

32. ECMA NLA
Ackerman D. J.
Network Layer Architecture
ECMA/TC24/81/15 - p. 1-10
January 81

33. ECMA NLP
Network Layer Protocol
ECMA/TC23/81/107
Augustus 81

34. ECMA NUISO
Notes on Using the ISO Reference Model of OSI
ECMA/TC23/80/46
March 80

35. ECMA SDFTP
State Diagram for File Transfer Protocol
ECMA/TC23/80/210 second draft
December 80

36. ECMA SP
Session Protocol
ECMA/TC23/81/100 p. 1-120
July 81

37. ECMA SCPR
Session Protocol Conformence Requirements
ECMA/TC23/81/157 p. 75-131

38. ECMA TC23/81/17
Brenner J. B.
Structure of the upper layers of OSI ECMA TC23/81/17
p 1 - 7
January 81

39. ECMA ADDRESS :
Rue du Rhone, 114 CH-1204 Geneve (Suisse)

40. GIEN 78
Gien M.
A file transfeer protocol
Computer networks, vol 2
p 312 - 319
1978

41. GRAY 79
Gray J.P.
Services provided to user of SNA networks
Datapro R.C. , CS50-510-301
p 301 - 307
September 80

42. GRAY NEIL
Gray & MCNeil
IBM Syst. Journ., vol 18, No 2
p 263 - 297
1979

43. HAL 79
Halsey, Hardy & Powning
public data networks : their evolution, interfaces, and status
IBM Syst. Journ., vol 18, No 2
p 223 - 243
1979

44. HB SNA
Honeywell Goes SNA
Datapro R.C., M17-111-081
August 81

45. IBM C
IBM Contribution to ECMA TC23 on OSI Architecture
ECMA/TC23/80/27
March 80

46. IIPDN
Grossmann, Hinchley & Sunshine
Issues in international public data networking
Computer Networks, vol 3
p 259 - 266
1979

47. INWG 96
Cerf, McKenzie, Scantlbury & Zimmermann
Proposal for an Internetwork End-to-End Transport Protocol
IFIP WG 6.1
p. H6-H25
February 78

48. ISO DTSS
Draft Transport Service Specification
ISO/TC97/SC16/WG6/Berlin N551
December 80

49. ISO MF
Classified OSI Management Functions
ISO/TC97/SC16/ N227
p. 41-181
October 80

50. ISO NS
ISO/TC97/SC16/ N551
November 80

51. ISO/TC97 RMDD
Detailed Description of the Resulting Architecture of OSI
ISO/TC97/SC16 N227
p. 41-181
June 79

52. KVR 80
Helbig P.
Klassification Verteilter Rechnen Systeme
Technisch Universitat Berlin KU-A1, Kurfurstendamm 202,
D-1000 BERLIN 15 Fed. rep. Deutschland
October 80

53. MACCHI 79
Macchi & Guilbert
Teleinformatique
Dunod Informatique
1979

54. MERCURY 78
Mercury J.Y. & Thierry L.
Principaux Concepts d' Architecture de Systemes Distribuès
Informatique et Gestion, No 100
p 89 - 95
October 78

55. NCR CA
NCR Comparative Analysis
NCR
January 81

56. OSI RM
Open System Interconnections Basic Reference Model
ISO TC97 - SC16 - DP498
January 81

57. POU 78
Pouzin & Zimmermann
A tutorial on protocols
Proceedings of the IEEE, vol 66, No 11
p 1346 - 1370
November 78

58. PPNI
Cerf V. G. & Khan E. R.
A Protocol for Packet Network Interconnection
IEEE Transactions on Communications, Vol. COM-22 p. 637-648
May 74

59. ROSI
Moldow B. D. IBM
Reality and The proposed OSI Standard
Data Communications p. 77-80
June 81

60. SKEES 81
Skees D. W.
Computer Software for Data Communications
Lifetime Learning Publications
1981

61. SNA CS
SNA Communication Subsystem for 9100/ATPF
NCR
1981

62. SNA FPRM
SNA Format and Protocol Reference Manual
IBM

63. SNA GI
SNA General Information
IBM GA27-3102-0 file No S370-09
January 75

64. SNA IACF
Introduction to advanced communications functions
IBM, GC30-3033-1
1979

65. SNA ILLS
Introduction to LU-Lu sessions
IBM
1979

66. SNA SPU
Services Provided to users of SNA Network
Datapro Research Corporation, CS50-510-301, p. 301-307
September 80

67. SUN DAL
Sunshine & Dahal
Connection management in Transport Protocols
Computer Networks, vol 2
p 454 - 473
1976

68. SUNSHINE 77
Sunshine C.A.
Interconnection of computer networks
Computer Networks, vol 1
p 175 - 195
1977

69. TANENBAUM 81
Computer Networks
Prentice Hall
1981

70. TDC 81
Bryan Gray
Trends in Data Communications
Systems International, p 39-42
September 81

71. TUC SP
Schindler & Burkardt
Structuring principles of the communication architecture of open systems
Technische Universitat Berlin

72. ROSI
Moldau B.D.
Reality and the proposed OSI standard
Data Communications

p 77 - 80
June 81

P R E M I E R E P A R T I E

ETUDE COMPARATIVE D' ARCHITECTURES DE RESEAU PAR RAPPORT AU MODELE DE REFERENCE OSI DE L' ISO.

-- CONTENTS OF PART ONE --

CONTENTS

OBJECTIVES

CHAPTER 1: NECESSITY OF AN ARCHITECTURE

1. Data communications and architecture . 1.1

2. Network control 1.2

3. Distributed architecture 1.3

4. Architecture and Standard Organisations 1.4

CHAPTER 2: THE REFERENCE MODEL

1. History and international Standardization bodies 2.1

2. ISO's Architectural model 2.3

2.1. History

2.2. General definitions : 2.5

2.2.1. Activity :

2.2.2. Entity :

2.2.3. System :

2.2.4. Protocols 2.6

2.2.5. Reference Model : 2.7

2.2.6. O.S.I. layering concept and structuring principles 2.9

2.2.6.1. Structure : 2.9

2.2.6.2. Fuller definitions : 2.10

2.2.6.2.1. (n)Service :

2.2.6.2.2. (n)Interface : 2.11

2.2.6.2.3. (n)Functions :

2.2.7. Protocols hierarchy operation

2.2.8. Levels of interconnection 2.13

2.3. Analogy : 2.16

2.4. OSI Reference Model general description : 2.18

2.4.1. Physical layer :

2.4.2. Data link layer :

2.4.3. Network layer :

2.4.4. Transport layer : 2.19

2.4.5. Session layer : 2.20

2.4.6. Presentation layer :

2.4.7. Application layer : 2.21

3. General considerations and OSI objectives 2.22

3.1. Controversy about the OSI goals 2.23

3.2. Gateways : 2.24

3.2.1. Types of gateways :

3.2.2. Remark : 2.25

3.3. Validity of the layering decomposition 2.27

3.3.1. Network layer

3.3.2. Transport layer 2.31

3.3.3. Higher layers 2.32

3.3.3.1. Controversy

3.3.3.1.1. Layers structure

3.3.3.1.2. Concluding remarks 2.35

4. Conclusions 2.37

CHAPTER 3: ARCHITECTURE DESCRIPTIONS

1. INTRODUCTION 3.1

2. STRUCTURING PRINCIPLES 3.2

3. SNA DESCRIPTION 3.4

3.1. Introduction to SNA

3.2. System cuts Appendix A

3.2.1. Network Addressable units (NAUs)

3.2.2. Network Configuration

3.2.2.1. Domain :

3.2.2.2. Host :

3.2.2.3. CUCN :

3.2.2.4. Cluster Controller :

3.2.2.5. Terminal node :

3.2.3. Sessions between NAUs :

3.2.3.1. Definition :

3.2.3.2. Session types :

3.2.4. End-User (EU) :

3.2.4.1. definition :

3.2.4.2. Different EUs :

3.2.5. Sessions establishment

3.2.6. SNA architecture

3.3. Service Cuts

3.3.1. SNA layering structure

3.3.2. Application layer

3.3.2.1. function :

3.3.3. Function Management Layer

3.3.3.1. Objectives :

3.3.3.2. Functions

3.3.3.3. composition :

3.3.3.3.1. NAU Service Layer

3.3.3.3.2. Data Flow Control

3.3.3.4. OSI equivalents :

3.3.4. Transmission Subsystem

3.3.4.1. Objectives :

3.3.4.2. Composition :

3.3.4.2.1. Transmission Control Element

3.3.4.2.2. Path Control

3.3.4.3. Data Link Control

3.3.4.3.1. Objectives

3.3.4.3.2. Services provided by the DLC :

3.3.4.3.3. OSI equivalent :

3.3.4.4. Physical Control level

4. DNA description 3.8

4.1. Introduction to DNA

4.2. System cuts Appendix A

4.2.1. Nodes :

4.3. Service Cuts Appendix A

4.3.1. DNA layering structure

4.3.2. Application Layer

4.3.3. Network Management Layer

4.3.3.1. Objectives

4.3.3.2. Functions

4.3.3.3. Composition (1):

4.3.3.4. Services provided to users

4.3.3.5. OSI eguivalent

4.3.4. Network Application Layer (N.A.L.)

4.3.4.1. Objectives

4.3.4.2. Composition

4.3.4.3. Services provided

4.3.4.4. Functions

4.3.4.5. OSI equivalent

4.3.5. Session Control Layer and Network Control Layer

4.3.5.1. Objectives

4.3.5.2. Functions and composition

4.3.5.3. Services provided to upper layer

4.3.5.4. OSI equivalent

4.3.6. Transport Layer (ISO network level)

4.3.6.1. Objectives

4.3.6.2. Functions

4.3.6.3. Composition

4.3.6.4. Services provided to upper layer

4.3.6.5. OSI equivalents

4.3.6.6. note

4.3.7. Data Link Control Layer (D.L.L.)

4.3.7.1. Objectives

4.3.7.2. Composition

4.3.7.3. Functions

4.3.7.4. Services provided to upper layer

4.3.7.5. OSI equivalent

4.3.8. Physical Layer

4.3.8.1. Objectives

4.3.8.2. Functions

4.4. Protocol cuts

4.4.1. Introduction

4.4.2. Application layer

4.4.3. Network Management Layer

4.4.3.1. Composition

4.4.3.2. NICE protocol

4.4.4. Network Application Layer

4.4.4.1. Data Access Protocol (DAP)

4.4.4.2. Services provided :

4.4.4.3. Dialogue exemple :

4.4.5. Network Service Layer and Session Control Layer

4.4.5.1. Network Service Protocol (NSP)

4.4.5.2. Services provided.

4.4.5.3. NSP messages types

4.4.5.4. Operation of a logical link and dialogue exemple

4.4.6. Transport Layer (OSI network layer)

4.4.6.1. General definitions

4.4.6.2. Routing Protocol

4.4.7. Data Link Control

4.4.7.1. Composition

4.4.7.2. DDCMP :

4.4.7.2.1. Description :

4.4.7.2.2. DDCMP messages types :

4.4.7.2.3. DDCMP operation :

4.4.7.3. MOP (1) :

4.4.8. Physical Layer

5. DSA description 3.13

5.1. Introduction to DSA

5.2. System Cuts 3.14

5.2.1. Host :

5.2.2. Network Processors :

5.2.3. Satellites :

5.2.4. Terminals :

5.3. Service Cuts

Appendix A

5.3.1. DSA layering structure

5.3.2. Application layer

5.3.3. Presentation layer

5.3.3.1. Objectives

5.3.3.2. Services provided

5.3.3.3. OSI equivalents

5.3.4. Session Layer

5.3.4.1. Objectives

5.3.4.2. Composition and functions

5.3.4.3. Services provided

5.3.4.4. OSI equivalents

5.3.5. Transport Layer

5.3.5.1. Objectives

5.3.5.2. Functions

5.3.5.3. Services provided

5.3.5.4. OSI equivalents

5.3.6. Network Layer

5.3.6.1. Objectives

5.3.6.2. Functions

5.3.6.3. Services provided

5.3.6.4. OSI equivalents

5.3.7. Data Link layer

5.3.7.1. Objectives

5.3.7.2. Functions

5.3.7.3. Services provided

5.3.7.4. OSI equivalents

5.3.8. Physical layer

5.3.8.1. Objectives

5.3.8.2. Functions and Services provided

5.3.9. Network Administration

5.3.9.1. Objectives

5.3.9.2. Services provided

5.3.9.2.1. NOI :

5.3.9.2.2. NAD :

5.3.9.2.3. NASF :

5.3.9.3. OSI equivalent

5.4. Protocol cuts Appendix A

5.4.1. Application layer

5.4.2. Presentation Layer

5.4.2.1. Composition

5.4.2.2. Standard Device Protocol

5.4.2.3. Transparent Protocol

5.4.2.4. Data Description Protocol

5.4.3. Session Layer

5.4.3.1. Composition

5.4.3.2. Connection Protocol

5.4.3.2.1. Services provided and functions performed

5.4.4. Transport Layer

5.4.4.1. System Communication Facility

5.4.4.2. Services provided

5.4.4.3. Operation and example of a Transport connection establishment

5.4.5. Network Layer

5.4.6. Data Link Layer

5.4.6.1. HDLC Lap-B Protocol

5.4.7. Physical layer

6. CNA DESCRIPTION 3.18

6.1. Remark :

6.2. Introduction to CNA

6.3. System cuts Appendix A

6.4. Service cuts Appendix A

6.5. Protocol cuts Appendix A

7. ARCHITECTURAL COMPARISONS 3.20

7.1. Global approach

7.2. Services Provided in reference to the OSI Reference Model 3.24

7.2.1. Physical Layer 3.24

7.2.2. Data Link Layer

7.2.3. Network Layer 3.25

7.2.4. Transport Layer 3.26

7.2.5. Session Layer 3.27

7.2.6. Presentation Layer 3.28

7.3. Advantages and disadvantages of those Architectures 3.29

7.3.1. SNA 3.29

7.3.1.1. Remark :

7.3.1.2. Advantages :

7.3.1.3. Disadvantages 3.30

7.3.2. DSA 3.32

7.3.2.1. Advantages :

7.3.2.2. Disadvantages :

7.3.3. DNA 3.33

7.3.3.1. Advantages :

7.3.3.2. Disadvantages :

8. CONCLUSION 3.35

OBJECTIVES

The first part of this work is devoted to an approach of what is the Open System Interconnection Reference Model (OSI RM), and to a contribution to a comparative analysis of some Network Architectures versus the OSI Reference Model.

Our goal is not to achieve a total description of the chosen architectures or of the OSI RM. In fact we aim :

- to point out some important concepts of the OSI Reference Model, explain them and give an overview of the Reference Model evolution and its impacts on the world of Network Architectures.

- to realize an overall description of four private Network Architectures in order to underline their relations to the OSI Reference Model and its influence upon their future developments.

Chapter 1: NECESSITY OF AN ARCHITECTURE

1. DATA COMMUNICATIONS AND ARCHITECTURE .

In the world of 1980 four major factors are present in the data processing and data communications :

- the need for data communication protocols that transcend the limitations of the older protocols (BSC , ...)
- the ability to move intelligence into smaller and smaller devices
- the development of communications among newer classes of devices .
- the need to interface into the newer common carrier environments and the capability to interconnect heterogenous devices .

Thus any product to be developed must be versatile enough not only to fit into the new environement , but also be able to adjust to new factors as they appear in the future. This means that each product should follow a set of standards that allow devices to be intermixed in a variety of ways.
In other words, a master plan - device independent - would be needed. This plan should contain all the allowed communication protocols between various types of equipments. A developer wishing to create a new product would thus provide those communication capabilities and respect these protocols in order to allow heterogenous interconnections.
A master plan or architecture would then make possible to use diverse products in different customer networks, Public Data Networks, in different parts of the same network, and/or at different time in the same network.

A communication architecture aiming to be the unifying force for all data communication products should specify :

- the logical concepts and structures
- a set of rules and guidelines
- a set of allowable network configurations

These aspects of the architecture , when combined in different ways, specify a particular product. Thus an architecture ensures that all communications related products , work together in a consistent and upward-compatible manner. The resultant network products , both those provided by a vendor and those developed by the user , can evolve to take advantage of new techniques and facilities with a minimum impact on the operational environement.

Within the overall aim of products cohesion, there are also some key goals that an architecture must satisfy:

- to make the network transparent to the end-user and application programmer.
- to better manage change in any of the network's elements.
- to enable multiple host systems or other intelligent devices to be connected to that same network.

An architecture is also intended to encompass all user networking requirements, ranging from simple networks of non programmable terminals through large networks of mixed terminals types and applications, and through all kinds of Local Area Networks, up to fully interconnected networks containing multiple hosts.

Further, the architecture should cover existing communication products and future products with equal facility by allowing selective omission and replacement of functional levels according to configuration rules and requirements.

2. NETWORK CONTROL

An architecture must provide a network control philosophy which integrates control completely into the network so that the network portion of the communication system does not rely on other components to keep it operational.
Also provisions for high efficient, high-reliable, low-overhead networks, extensible and ultra-resilient networks, and data-secure networks must be inherent in the architecture.

3. DISTRIBUTED ARCHITECTURE

The architecture must facilitate the distibution of processing capability and control capability throughout the communication system. This is accomplished through a logical partitioning of the communication system into a set of network functions and a set of two or more processing functions that communicates via the network functions. These processing functions being treated identically whatever their size.

This distribution of intelligence allows:

- local control : control over specific application functions and data bases can be held, (where such control is meaningful) in the local processing environment.

- line savings - local terminals or hosts no longer need to make so many cross-network accesses to other processus,with subsequent savings in line utilisation and line charge.

- improved availability - local processing does not need to depend on the availability of remote resources. If these remote resources are unavailable, local processing can continue in a degraded mode until the remote resources become free again.

- application oriented terminals - terminals can be constructed or adapted for particular applications. A user can tailor the processing to his own requirement.

4. ARCHITECTURE AND STANDARD ORGANISATIONS

Most major computer vendors have developed a cohesive communication architecture of some sort (fig. 1).
At the same time some of the major international organizations are also developing their own architectures. Perhaps the most important of these is the ISO's(1) Open System Interconnection (OSI) model that we intend to study in the following.
The objectives of those architectures are similar : "to support data communications between devices and users"; but the ways followed are dissimilars.

vendor	Architecture
Burroughs	Burroughs Network Architecture (BNA)
Digital Equipment	Digital Network Architecture (DNA)
Honeywell	Distributed systems Architecture (DSA)
IBM	Systems Network Architecture (SNA)
Univac	Distributed Communication Architecture (DCA)
ISO	Open Systems Interconnection (OSI)

Figure 1 : vendor Architectures

(1) - I.S.O. : International Standard Organisation

Chapter 2: THE REFERENCE MODEL

1. HISTORY AND INTERNATIONAL STANDARDIZATION BODIES (fig. 2)

The growth of digital data transmission systems and intelligent networks causes a need for standards. Standards are required to ensure one piece of equipment, be it a host computer or user terminal, can communicate with another piece of equipment. Without standards, mixed vendor networks or public networks should not be possible; international communications should be difficult; and complex homogenous networks would be very hard to implement.
The standards must consider a number of physical characteristics as well as the operational procedures necessary to facilitate the data transfer. The standards developed by various governmental agencies, industry groups, study groups and manufacturers not only govern the kind of equipment the user buy, but also govern most aspects of its operation. The rules among other things,determine how fast the equipment operates, the type and quality of carrier facilities the user may access, compatibility of equipment with these facilities, with equipment from other manufacturers, and the types of codes and protocols that must be followed.

1. I.S.O.
   "The primary international body concerned with world-wide data communications standards is Technical Committee 97, subcommittee 6 of the International Standards Organisation (ISO). Membership in ISO is limited to recognized standard-making bodies. There are now 62 full members of ISO and 19 correspondant members from developing countries that do not have standard-making bodies of their own. The job of ISO TC 97/SC 6 includes constant appraisal of data communications through all types of telecommunications media, as well as the ability of its members to provide themselves with data communications to link data processing systems."

2. CCITT :
   "Also working in international data communication standards-making area is the International Consultative Committee on International Telegraph and Telephon (CCITT), a permanent organ of the International Telecommunication Union.
   The CCITT weights all technical, operational, and tariff matters at the governmental level for all types of information transfer."

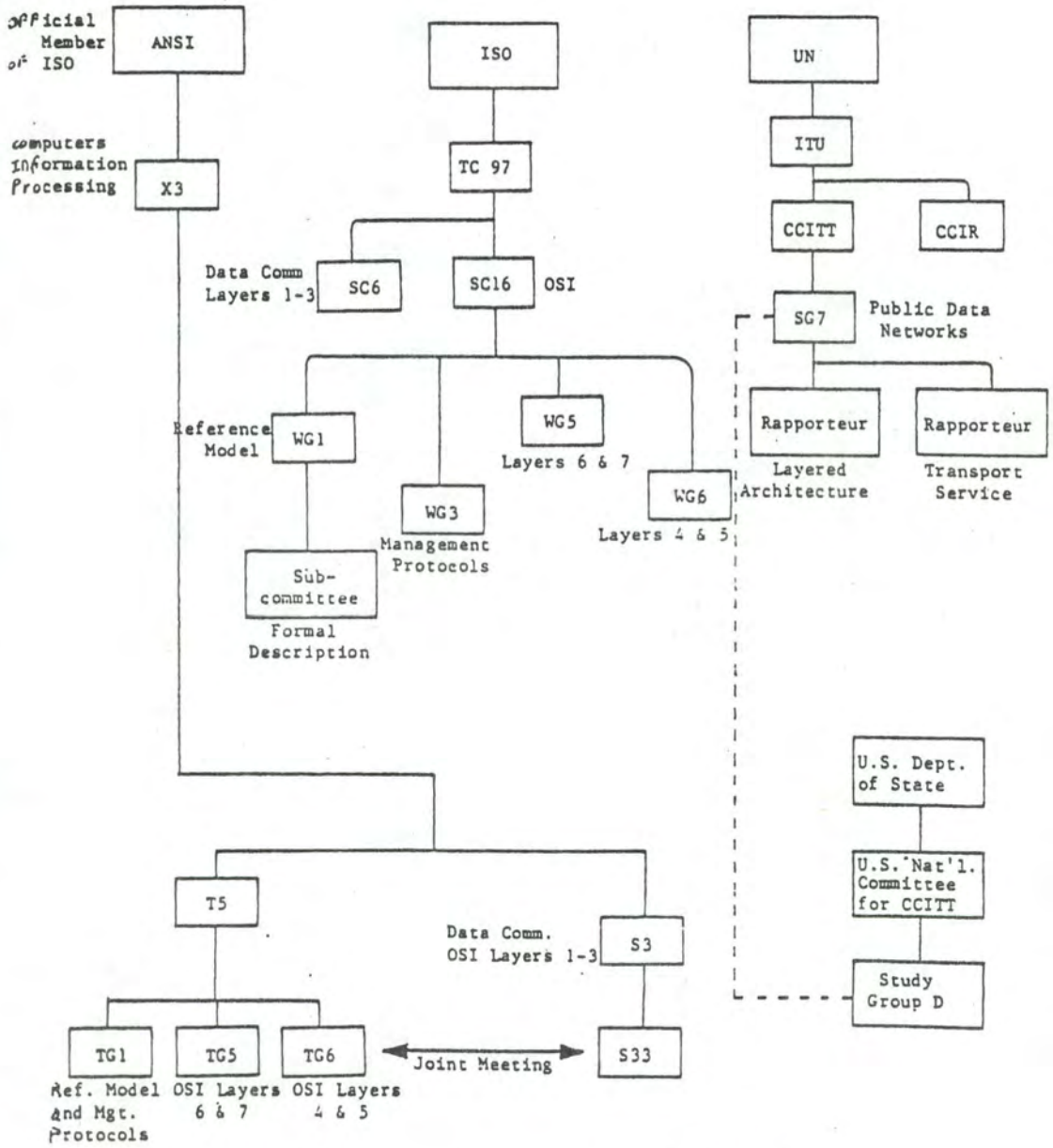


Figure 2 : International Standard bodies

3. ANSI :

The mission of subcommittee X383 of sectional Committee X3 on Computers and Information Processing of the American National Standards Institute (ANSI) is to define the characteristics of digital data-generating and receiving systems. It is responsible for developing and recommending standards for data communications. ANSI standards deal with the quality and characteristics of data during transmission.
It is also the U.S. representative to the ISO, and it will often either approve international standards for use in the U.S. or tailor ISO proposals to the needs of american industry.

2. ISO's ARCHITECTURAL MODEL

2.1. History

It is the long-term objective of ISO to develop full intersystem compatibility between the various products and services offered by vendors and PTT's worldwide. The first step was the development of an architectural model for the purpose of providing a global architecture solution. This model has been named the Open Systems Interconnection (OSI) and was created by a new subcommittee (SC16).

The standardization effort, which began in 1977, has made significant progress. Internationally, there are now three organizations participating:

- The European Computer Manufacturers Association (ECMA).
- I.S.O.
- C.C.I.T.T.

A number of national or international professional standards bodies provide much of the support and technical input to the international activities.

Initially the work was centered around development of an architectural model which would be used to identify the functional requirements, establish a structure for orderly progress, and allow identification of areas already standardized, areas under development, and areas which need to be developed to provide a set of standards for OSI. Now the OSI work has progressed to the point where the reference model has reached a more precise and stable definition. Committees are being formed to begin development of detailed standards as a result of work items identified by the reference model.

As ISO defines it :

> " OSI refers to standards for the exchange of informations among terminal devices, computers, people, networks, processes etc .., that are 'open' to one another for this purpose by virtue of their mutual use of the applicable standard.
>
> Openness does not imply any particular systems implementation, technology, or interconnection means, but rather refers to the mutual recognition and support of the applicable standards." [OSI 80]

The practical impact of OSI is not far off. The U.S. National Bureau of Standards (NBS) is basing its standards for federal procurement on the OSI model. Its guidelines have been published in the federal register in 1981 and by 1982, the agency intends to complete a series of standards based on ISO. In Europe the ECMA bases its standards and recommendations on OSI. And internationally, the OSI Reference Model has also been chosen by the CCITT as its model. To this end, parts of the OSI have been and are being implemented by many PTT´s worlwide (France, Belgium,....).

A controversy exists meanwhile about the real impact of OSI and its future developments. Some weaknesses can be pointed out in the current OSI architecture and the OSI goals which should have to be faced.
This subject will be discussed in section 3.

2.2. General definitions :

Before the detailed description of the composition of OSI model, we will specify the concepts of activity, entity, system, reference model, layers, and levels of interconnection in a network architecture.

2.2.1. Activity :

One or several processes [POU 78], along with appropriate resources, which work towards a common goal, may be called an activity.

2.2.2. Entity :

Anything capable of sending or receiving information. It may be a process, a terminal, a program in a computer, a file, an activity,...

2.2.3. System :

In the concept of OSI, a system is a set of one or more computers, the associated software, peripherals, terminals, human operators, physical processes, information transfer means, etc.., that forms an autonomous whole capable of performing information processing. OSI is concerned with the exchange of information between systems and not the internal functioning of each individual system.

In other words, the Reference Model of Open Systems Interconnection constitutes the framework for the development of standard protocols for communication between two peer layers located in separate pieces of equipment. Formats and protocols for adjacent-layers communication within a system would not and need not be standardized.

But as it is said in the basic OSI Reference Model :

"OSI is not concerned only with the transfer of information between systems , i.e. in communication but also with their capability to interwork to achieve a common (distributed) task . In other words, OSI is concerned with cooperation between systems , which is implied by the expression systems interconnection".

The scope of the cooperation among open systems entails a broad range of major subjects of which the following have been identified :

1. interprocess communication, which concerns the exchange of information and synchronization of activity between application-processes;

2. data representation which concerns all aspects of the creation and the maintenance of data description and data transformation for reformatting data exchanged between systems;

3. data storage, which concerns storage media, and file and data base systems for managing and providing access to data stored on the media;

4. process and resource management which concerns the means by which application-processes are declared, initiated and controlled, and the means by which they acquire resources;

5. integrity and security, which concerns the definition, compilation, linking, testing, storage, transfer, and access to the programs executed by application-processes, or the data handled by these programs.

2.2.4. Protocols

All communication is governed by rules of procedure. The most useful kind of communication, wether between people or machines, is a dialogue(two-way-interconnection) rather than a monologue. Conversation cannot avoid having procedural rules to determine when the listener may speak again, what to do if the message is not understood, etc... .

Between people, these rules are informal, but subtle and complex. Between computers, they have to be formal, accurate, and more simple than for humans.

(Because of the need for precision and formality, very difficult problems arise in designing, testing and verifying the correctness of protocols. To cope with them a lot of formal techniques are growing, having their own advantages and disadvantages.)

Thus, two processes runing in different locations must

use the exchange of messages to coordinate their actions and achieve synchronization. This message exchange follows carefully designed procedures called Protocols.

A Protocol can thus be defined as

> A set of rules or conventions (semantic and syntactic) governing the cooperation of components achieving some kind of activity. It defines then the relations and behaviour between those components, and determines the communication behaviour of peer-entities in performance of functions.

Other definitions are found throughout the litterature, but have similar meanings.

The main characteristic of protocols is the ability to work where the timing and sequence of events can be unknown and where transmission errors are expected.

Another way to define protocols would be the following :

- Protocol functions are accomplished by the exchange of message between processes. The format and meaning of these messages form the logical definition of the protocol.
- Rules of procedure determine the actions of the processes cooperating in the protocol. The set of these rules constitutes the procedural definition of the protocol.

Protocol definition is then :

> "The logical and procedural specifications of the communication mechanism between processes. The Logical definition constitutes the syntax , while the Procedural definition forms the semantic of the protocol."

2.2.5. Reference Model :

> In general, a model is an abstraction or simplification that makes a concept more understandable.

In order to comprehend models of complex systems, it is important to partition the structures into easily understood parts. Communication systems are often envisioned in terms of layers of functions and layers of services. These layers being represented, for convenience, in a vertical sequence.

The rationale for structuring system into layers is to provide a convenient partitioning of functions which will ensure :

- Independance of activities between layers (a change in one layer will only affect that layer).
- Information hiding of specific implementation of each layer as seen by users of that layer.
- Sharing of common services by different applications when it is usefull and possible.
- Sequentiality of events in time from layer to layer.

The purpose of a reference model is to provide a complete reference structure for the total universe of standards necessary for interconnection. It will allow to :

- Position existing standards into perspective
- Identify areas requiring additional developement
- Identify areas where standards should be developed
- Expand without disrupting previously defined protocols and interfaces
- Be subject to formalization and modeling for determining completeness and correctness of functionality.

The model of OSI, that we will refer to as the OSI Reference Model, matches these principles.

2.2.6. O.S.I. layering concept and structuring principles :

2.2.6.1. Structure :

The basic structuring technique for building the architecture of the OSI Reference Model consists thus of a hierachical assembly of layers, as shown in fig.3. It can be more formally defined as follows :

1. Layer (n) of the structure makes use of type (n-1) services provided by the lower layers through the type (n-1) access.

2. The structure of these layers is not known by layer (n) which considers only the services provided by a type (n-1) box.

3. Layer (n) is made of (n)entities which cooperate according to a type (n) protocol. These (n)entities are also called peer (n)entities or peer-processes.

4. The type (n)entities perform type (n)functions using the type (n-1) services to provide type (n) services to layer (n+1) .

5. Data and control information exchange between two adjacent layers is made through an interface.
   - The interface defines which primitive operations and services are provided to the upper layer by the lower one.
   - Control information aims to the correct operation of the interface and to support tasks requests and reports (about progress or completion).
   - Data passing across the interface are messages associated with the protocols and destined to the remote corresponding layer.

6. Data and control information exchanged between two peer-entities is made through a logical or virtual link. This logical link is implemented via the lower supporting layers.

The layers must be chosen so as to strike a balance between the cost of complexity and the benefits of subdivision. It may be difficult to prove that any particular layering selected is the best possible solution. There are many conflicting factors affecting the choice of layers. They include desires for modularity, subdivision,

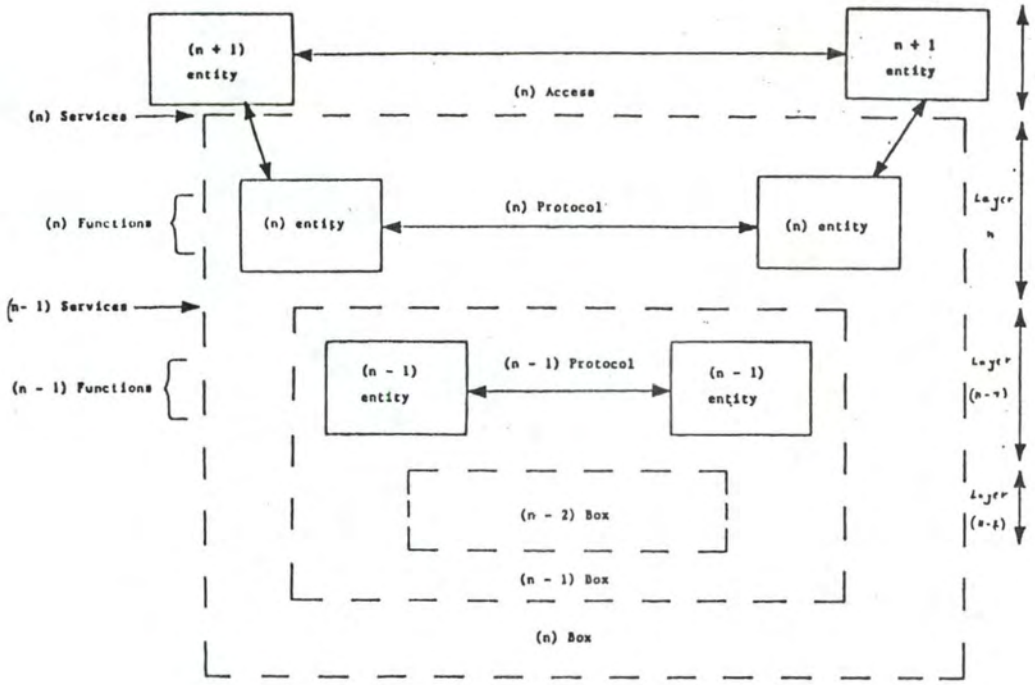


Figure 3 : Hierarchical Layers

overall simplicity and coherence.

The specification of a layer of the architecture must also in some way refer to the set of services provided by lower layers. This will be done by making use of access functions which define the way (how,when,...) these services are accessed.

The set of access functions to a service should be viewed as a means to describe the logical structure of the network. It does not necessary imply the existence of the corresponding interface in any implementation of a piece of network. This definition no more implies a one-to-one correspondance between type (n) and type (n-1) entities. For instance, a type (n-1)entity can serve several type (n)entities.

[for more information, look at [OSI RM] p 9]

2.2.6.2. Fuller definitions :

2.2.6.2.1. (n)Service :

As defined in the OSI Reference Model, a (n)service is the capability of the (n)layer provided to the (n+1)layer Entities at the boundary between the (n)layer and (n+1)layer.

In other words, the services of a layer are provided to the next higher layer, using the functions performed within the layer and the services available from the next lower layer as said above. An entity in a layer may provide services to one or more entities in the next higher layer and uses the services of one or more entities in the next lower layer. The services referred to are the set of properties and assumptions that can be relied upon by the communication user when designing the way in which its higher level activities will be performed. The definition of a service is, in this context, concerned with the expected properties of the communication. It must be distinguished from the specification of interfaces within a system which allow access to the communication components, and contains many local implementation choices that do not need to be standardised.

The service allows its users to establish a connection leading to the party with which they

need to communicate. The definition of the service involves the definition of those aspects of the set of actions on a stimuli from the connection which form an essential part of the communication. [CHO 80]

2.2.6.2.2. (n)Interface :

We could define it as the set of access functions needed by/available in the (n+1)layer to access/use the services provided by the (n)layer.

2.2.6.2.3. (n)Functions :

Could be defined as the set of functions or tasks performed by the (n)layer entity(ies) to provide a (n)service(s) to the upper layer.

This set may vary in size, according to the quality of the services provided by the lower layer, in order to provide a requested or offered service to the upper layer.

2.2.7. Protocols hierarchy operation

The layer (n) in a (n)system then carries on a conversation with layer (n) in another (n)system.

- The rules and conventions used in this conversation are collectively known as the layer (n) Protocol (fig 4).
- The set of layers and protocols is called the network Architecture.
- When considering a layer, it is the peer entities or peer processes that communicate using the layer protocol. But have in mind that, in reality, no data are directly transferred from layer (n) in a (n)system to layer (n) in another (n)system -except in the lowest physical layer-. Instead , each layer passes data and control information to the layer imediately below it, until the lowest layer is reached.
- At the lowest layer, there is a physical communication between systems (opposed to the virtual communication used by the higher layers or higher entities).

F4
F5

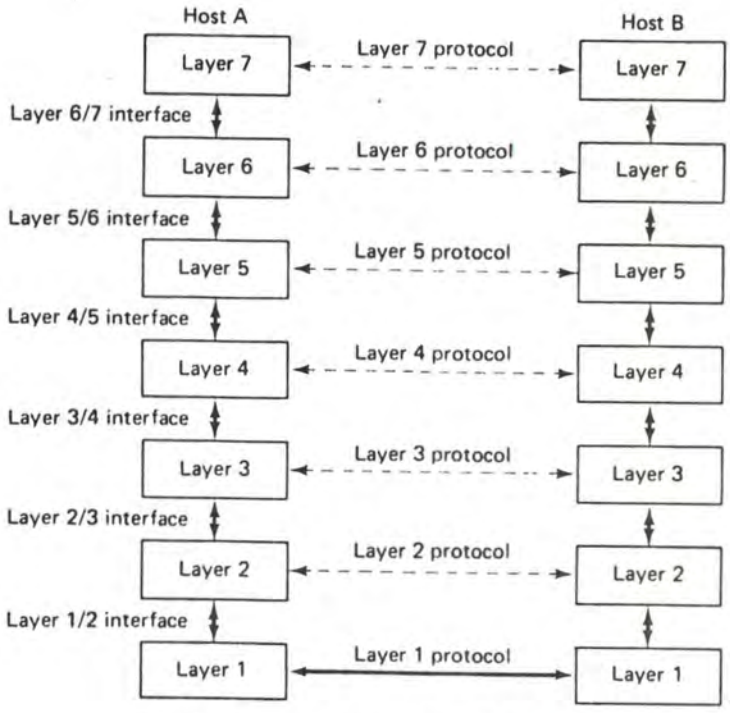


Fig. 4: Layers, protocols, and interfaces.

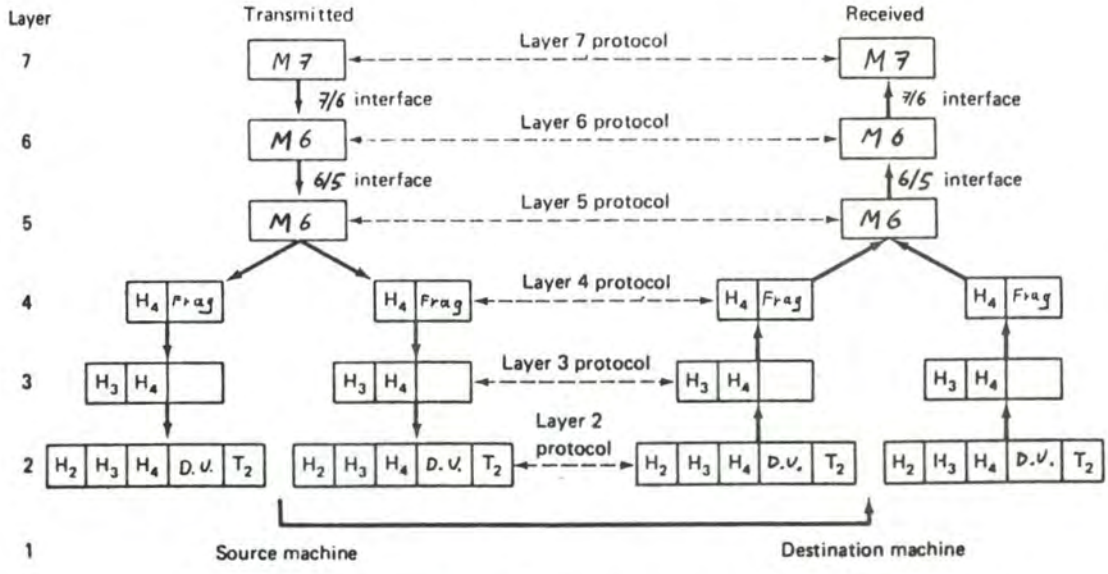


Fig. 5: Actual information flow supporting virtual communication in layer 7.

- Between each pair of adjacent layers there is an interface. This interface, as we said it before, defines which primitives operations and services the lower layer offers to the upper one. Those services are accessed through access functions that must be cleanly specified, and, aim to minimize the amount of information passed between layers and to help to a modular designing making simpler (and transparent) modifications in one layer.

To better explain the operation of this multilayer communication we will give two exemples.

1. Imagine two persons - peer processes or peer entities in the highest layer 1 - , one in Belgium, the other in USA, who want to communicate. Since they have no common language they each engage a translator - peer entities at a lower ajacent layer 2 - to provide them with a translation (presentation) service, each of whom, in turn, contacts an engineer - peer entities at the lowest layer 3 -.

   The man in Belgium wishes to convey a letter to his US peer. He writes it, in french, and passes it across the 1/2 interface to his translator who renders it as a formatted letter in another language (english, german, morse, ...) depending on the layer 2 protocol. The translator then gives the letter, through a 2/3 interface to his engineer for transmission by telegram, telephone, computer network, postal service, etc.., depending on what the two engineers have agreed on.
   When the message arrives, it passes to the translator through the 2/3 interface, is translated in some language, and passes through the 1/2 interface to the US man.

   The language used by the two peer-translators can be whatever they have agreed on before (morse code, vigenere cipher,...).

2. Consider now the seven-layer network architecture (fig. 5):

   A message M7, produced by a processing entity in layer 7, is to be sent, through a virtual support, to another processing entity. It will follow this way :

   - the message is passed to layer 6 according the definition of the layer 7/6 interface.

- Layer 6 transforms the message in a certain way (not always needful, but, for instance, assume a 'code translation') and passes the coded message, M6, to layer 5 across layer 6/5 interface.

- Layer 5 simply regulates flow and data exchange to maintain synchronization with distant peer-layer (virtual communication) and to avoid overloading of the upper layer. It passes then the message, M6, decomposed in fragments, to the layer 4 at a certain rate.

- Layer 4 breaks up the incoming fragments into smaller units adding a header to each unit (header contains needed information to allow destination layer 4 to reconstitute the fragment in correct order).

- Layer 3 , receiving those units, decides which outgoing media to use, adds its owns headers to the units received from layer 4, and passes the data to layer 2.

- Layer 2 adds header and trailer to each data unit received from layer 3 (in order to allow error trapping and correction) and passes the resulting unit to layer 1.

- Layer 1 which handles physical connections (electrical, optical, ...) transmits the received units, transformed in electric signals to an adjacent system layer 1. And hop by hop, the signals finally reach the destination system.

- There, the units move then upward, from layer to layer, with headers being stripped off as they progress. The message, M6, is reconstituted in layer 6 and passes finally as M7 to layer 7.

2.2.8. Levels of interconnection

Another characteristic common to the network architectures is that of having three levels of connection within the network. Figure 6 illustrates these three levels.

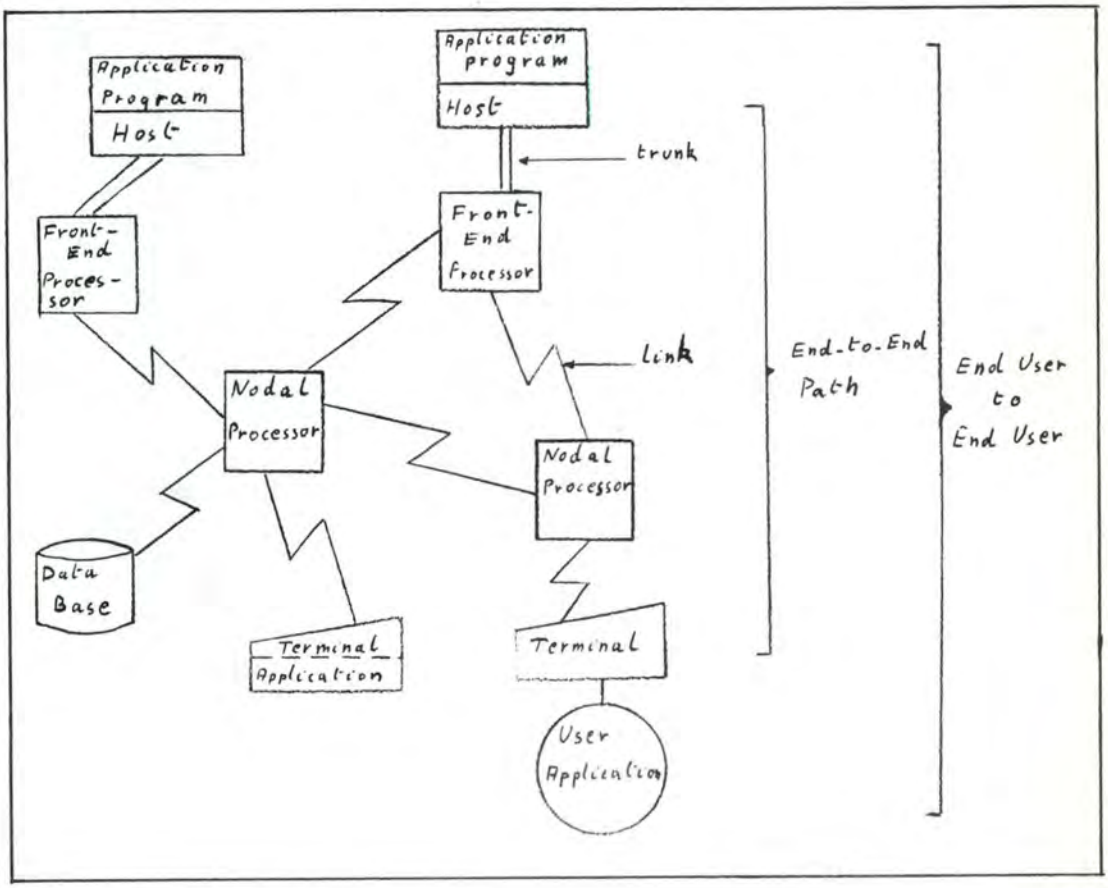


Figure 6 : levels of interconnection

1. The first or more basic level is that of the link between two nodes. This link could be a single communication line interconnecting two devices, or it might be a multi-dropped line managed by a communications Front-End, a remote concentrator, or even an intelligent controller. In any event it is the particular communication facility that interconnects two or more physical network entities. The connection at this link level is usually implemented via one of the High Data Link Control protocols (HDLC, SDLC, BSC, DDCCP) which rule the exchange of information between two adjacent nodes.

2. The second level of connection within the network is

that between originating node and destination node, and is referred as the end-to-end path, the Logical link or the virtual link. This is a logical connection or set of routing rules to move data from the input node to the target node.

- The physical path may be constant during the communication or it may vary dynamically, depending on the architecture.
- As a reference point, the communication standard X25 -the procedure for interfacing Public Data Networks(PDN's)- applies to the path level of network connections. It employs a manner of logically identifying a destination in the network, which some intelligence, within the network, translates into a physical address. It rules the communication between the originating node and an entry node to the PDN at the link level, and includes instructions so that the network can act at the path level to route data to its destination. The communication (data) then emerges at an appropriate exit node on the network and proceeds to its destination node via a link level connection. The end-to-end connection terminates at the destination node, which is also part of the transmission system.

3. The third level of connection is that between paired End Users which are always sources or sinks for data.

   One of the paired End User will usually be a person at a keyboard, or an application program, a card reader, printer, punch, or CRT screen. The other member of the pair often will be an application program receiving input from and/or sending output to the first. However, both could be application programs. An application executing in an intelligent controller may communicates with an application program in a host to retrieve data not present in its own storage, or to obtain new informations, for exemple.

   The End User is usually considered to be outside of the network architecture. Thus the network puts no restrictions on the functions within the End User or on the format of the interface to the End User. The End User and its interface is not architectured and is product dependent.

2.3. Analogy :

To see the role the reference model play in the development of products, assume that a cassette recorder and a radio are to be connected together.
"One way to do this is to study the electrical diagram of both devices, and try to determine appropriate points where the two devices might be electrically coupled so that sounds received by the radio are recorded on the cassette. It may be difficult to locate these appropriate coupling points, if the diagrams are not well documented. Electrical signals may also require some adjustment.

Interconnecting a cassette recorder and a radio became much easier if both manufacturers have anticiped the need, and installed some pluging sockets carrying well-defined electrical signals. In this case, the inner working of both devices do not have to be studied. Only the pins of the sockets have to be understood. These pins are the only needed view of the device. They can be called visibility points.

Computers systems are more complex than cassette recorders. Thus, interconnecting them require more than the definition of simple sockets. Computers contain a hierarchy of functions (applications - operating system - access method - ...). These hierachy of functionalities are called layers. Interconnecting layers requires the definition of protocols, which are the set of rules followed within each layer by the interactions between interconnected systems. Furthermore, protocol definitions assume some interaction between layers within each computer system. For example, an application program passes certain parameters to a Presentation management layer which will act according these parameters.
The set of rules followed by two adjacent layers when engaged in interactions between each other is known as an interface.

Once interfaces and protocols are defined between interconnected systems, they may work in cooperation to achieve a common purpose. They become a distributed system. If there exists a common set of definitions of all interfaces and protocols used in distributed systems, it is no longer necessary to study the inner working of each system. Interfaces and protocols are analogous to sockets in the connection of the cassette recorder and the radio set. They make interconnection much easier.

But if the development process required that all interconnected systems rigidly follow a model precisely defining systems in their entirety, then the developed products would only comprise computer systems of the same family.
Indeed, not many vendors would accept being constrained to such an extent in the design of their own products. but if the constraints are limited only to interfaces and protocols necessary for building a model, it becomes commercially

attractive for many vendors to accept these constraints as the counterpart of a wider market.
It can be said that a system conforms to a Reference Architecture when it contains the visibility points of the distributed system model, i.e., the interfaces and protocols for interconnection." [exemple out of "a tutorial on protocols",(1)]

These notions being defined, let's go back to the overview of OSI model.

---

(1) - "tutorial on protocols" : Pouzin - proceedings of the IEEE - nov 78 - vol 66 - n. 11 -

2.4. OSI Reference Model general description :

The OSI Reference Model consists of the seven following layers (as shown in fig. 7).

2.4.1. Physical layer :

The physical layer provides the mechanical, electrical, functional, and procedural characteristics needed to establish, maintain, and release physical connections (data circuit) between the device (Data Terminal Equipment or DTE) and the network (Data Circuit Terminating Equipment or DCE) or between two devices (DTE's).

2.4.2. Data link layer :

The Data Link Layer provides link connection, addressing, sequencing, and error recovery at the link layer. There is a link address that uniquely identifies a link connection in the link layer.

The existing link connections are point-to-point or multipoints.

2.4.3. Network layer :

The Network Layer provides control between two adjacent nodes. A point-to-point or network connection is supplied.

- One or more network connections may map to the same link connection.
- each endpoint of a newtwork connection is uniquely identified by a network address.

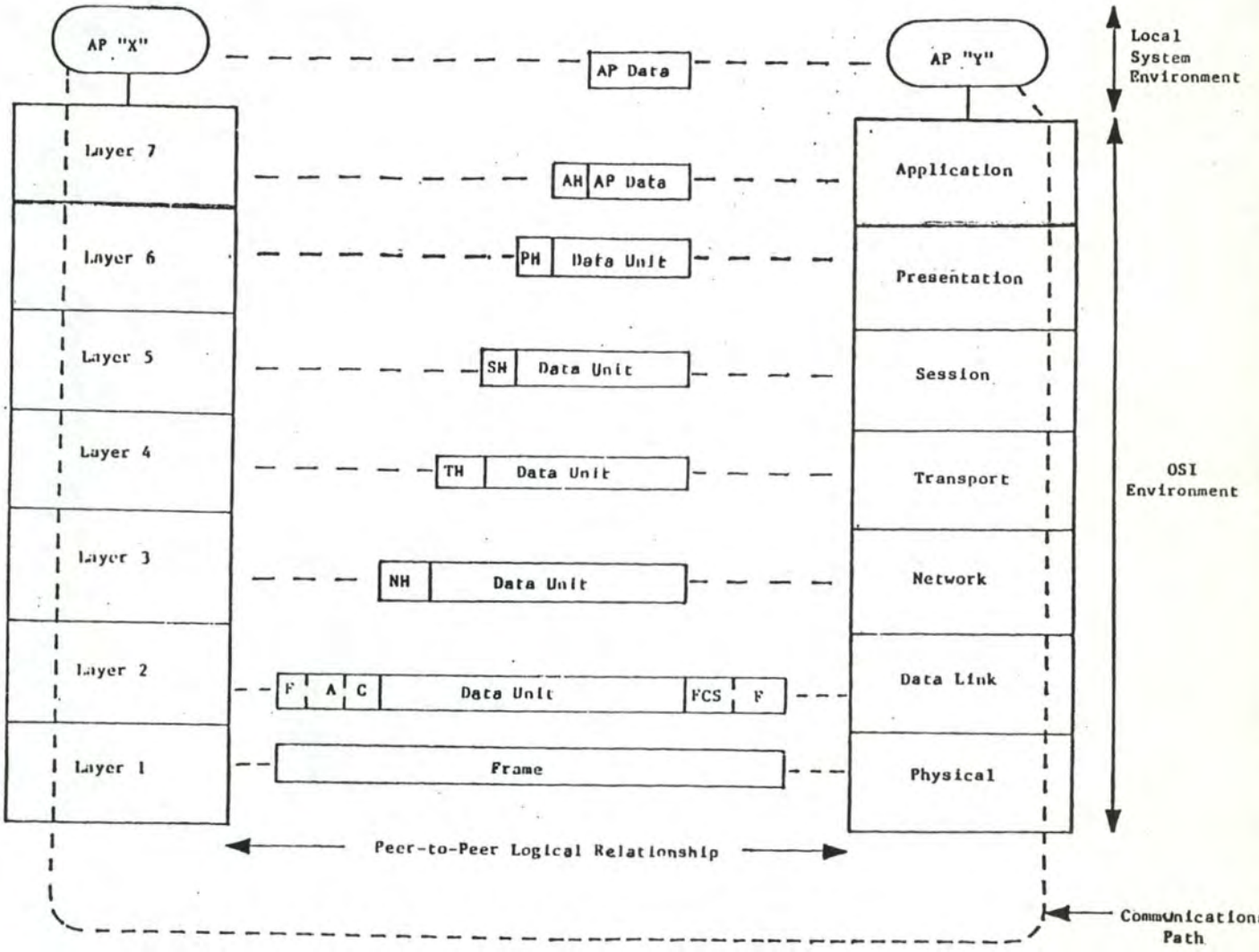


Figure 7 : ISO Model

- The functions provided by this layer include message routing, error notification and correction, and, optionally, segmenting and blocking.

Network layer facilities may be viewed as those directing the control of switching points, rather than providing for the transfer of data between the switching points.

Those three first layers rule communication between adjacent systems and have no end-to-end signification. On the other hand, the four upper layers are end-to-end or source-to-destination layers. The protocols are carried out by the end-systems, whatever the distance and path structure between them.

2.4.4. Transport layer :

the Transport Layer exists to provide a network independent interface to transport services users.

1. It provides control from user node to user node across the network. It provides a virtual end-to-end link between two upper layer communicating entities. Each virtual link has its own flow control.

2. This layer offers the functions of class-of-service selection, sequencing, end-to-end flow control, and expedited message transfer.

3. It relieves the session level from any concern with the detailed way in which the transfer of data is achieved (circuit switching, packet switching or datagram, X21, ...).

4. A transport connection is identified by a transport endpoint identifier.

5. One or more transport connections may map to the same network connection. This is called the upward multiplexing.

Layers 1 through 4 form the Transport Subsystem.

2.4.5. Session layer :

The Session Layer provides the support for the interactions between cooperating Entities of the presentation layer. The functions of the session layer may be classified in two categories :

1. binding and unbinding of sessions between two Presentation layer entities (Session Administration service) and
2. control of data exchange, i.e. synchronizing, delimiting, and recovery of data operations between two Presentation Layer Entities (Session Dialogue Service).

A session is identified by the session-endpoint-identifiers. Three types of interactions are defined :

1. two-way simultaneous (TWS)
2. two-way alternate (TWA)
3. one-way

2.4.6. Presentation layer :

The purpose of the Presentation layer is to provide the set of services which may be selected by the Application layer to enable it to interpret the meaning of data exchanged between communicating Application-entities (resolving syntax differences, handling format differences, assuming encryption if requested.
The model identifies three exemples of Presentation Layer protocols:

1. Virtual Terminal Protocols
2. Virtual File Protocols
3. Job Transfer and Manipulation Protocols.

Code translation would also be included in level 6.

The presentation service is location-independent and is considered to be on top of the Session Layer which provides the services of linking a pair of presentation entities.

2.4.7. Application layer :

The Application Layer is the highest layer in the OSI architecture. Protocols of this layer directly serve the End User by providing the distributed information service appropriate to an application, to its management, and to system management of OSI comprised those functions required to initiate, maintain, terminate, and record data concerning the establishment of connections for data transfer among application processes. The other layers exist only to support this layer.

An application is composed of cooperating application processes which intercommunicates according to application layer protocols. Application processes are the ultimate source and sink for data exchanged. A portion of application processes relate to application protocol, another to the system management, and the rest of the application processes is considered beyond the scope of the OSI Reference Model.
[adapted from OSI RM].

3. GENERAL CONSIDERATIONS AND OSI OBJECTIVES

The development of OSI standards is a very big challenge, the result of which will impact all future computer communication developments.

The goal of this effort is the linking of all terminals and computers in an organization, such that an authorized user could access any application or file regardless of its location and the product in which it resides.
There is one major difference between the ISO effort and the various computers vendors efforts. Whereas the computer vendors limit the equipment option to their own or, in some cases, to limited competitive products (or product lines), the ISO group wants to open the approach to any devices that implement certain standard interfaces and protocols. Further, in contrast to the computer vendors architectures, which are oriented to intra-company networks, the ISO approach will facilitate information exchange both within and between organizations of all kinds. As such, it could greatly accelerate the developments in such areas as distributed processing, international data communications, and office automation.

The OSI Reference Model is an abstract, a logical and functional description. It is not a system architecture nor a system implementation plan.

The objective of ISO for OSI is to standardize the protocols used for peer-to-peer communications (fig. 7); that is, for control coordination between equivalent layer entities in different user system. This means the semantics (type of data) and syntax (format of data) is standardized.
The interface between adjacent layers would have only loosely identified boundaries. The semantics, but not the syntax, would be standardized. Vendor's A products would be able to talk to vendor's B products if they both can respond to the same peer layer protocols at same levels. This does not mean all products will be able to communicate; because there is a universe of different functions off which various types of products use diverse subsets. Different products to be similar would have to implement similar function irrespective of the manufacturer.

Presently, only layers 1 through 3 have been functionally completed after nearly four years of network operational experience. One implementation example is the packet switching X25 standard (of which many not fully compatible releases exist).
The remainder layers (4-7) are still under development. But, construction of the higher-level standards is moving rapidly; this fact cannot be ignored by designers. The preliminary standards should be codified in the next several years. Further, these standards are being backed by the most important standard-making bodies - ISO, ANSI,

CCITT, ECMA -. The PTT's of various governments are also accepting these standards. This means products to be designed and sold to these bodies must meet the standards.
Failure to meet them means the products cannot be marketed to these bodies.
Evenmore, various consumers and users induce to the respect of these standards, in order not to lock themselves in a limited network environment, and look to the products providing those capabilities.

So the question arises of how the vendors intend to meet these standards (Honeywell has taken the approach of adaptate OSI Reference Model for the description of its communication architecture); and how vendors that have previously existing architectures react and migrate to respect the new standards.

The following points will be developed now: The first one sums up controversial ideas about OSI goals, approching the gateway conception of the OSI Reference Model. Finally the question of the validity of the layering decomposition of the OSI Reference Model will be considered.

3.1. Controversy about the OSI goals

To other people, the OSI Reference Model does not constitute a set of standards which, once applied, would insure for compatibility, but a Functions classification or a common view about what are communication functions, a means to obtain a standardized terminology in order to further develope standards.

OSI Reference Model should not evenmore be viewed as a complete set of the functions an architecture has to provide (forbidding extra functions to be implemented), but as a Reference Model that can be adapted to user requirements, if respected in what concerns its concepts and basic description.

More, OSI Reference Model is introduced in a world of already existing communication architectures and implementations. It is doubtful that the constructors will review their models and modify them to satisfy to the OSI Reference Model recommendations. It is more probable that they will provide some device, architecture of which will follow the OSI one, constituting an interface, a gateway to ISO universe.
The existent network architectures and implementations will thus continue to grow; but compatibility will be assured by OSI

gateways.

Considering OSI Reference Model, we could say that it ignores gateways, facing only open systems providing an output port; but we could say that OSI copes with the Gateway principles (OSI being concerned with cooperation between systems) and aims to define interface structures through which different systems are able to intercommunicate in a compatible manner.

3.2. Gateways :

Since different network types exist, it is logical to assume that users would want to use these different types within the same user system. (for instance an X25 PDN, a vendor network, and a Local Network). Therefore, the networks must be able to be interconnected somehow (fig. 8). The motivation for interconnecting networks is to provide one or more consistents services to users of the interconnected networks. To provide these services, either new end-to-end services protocols must be defined, or the service protocols of the individual networks must be made to internetwork. In either case, the issues of addressing, routing, buffering, flow control, error control, and security, must be considered. To reach this goal, a common denominator must be found. In this case, the ISO OSI model could provides one, achieving the description of an interface common to the existing architectures.

Another major objective of OSI is to minimize the impact of conversion while promoting the ability to migrate. There is a large variety of systems with different internal protocols, formats, and operation procedures. Within the context of Open Systems Interconnections, the solution is to promote and use gateway systems.

A Gateway is a device to which two or more networks are connected, and whose function is to convert one or more levels of protocols to other protocols, or in other words, to translate the dialogue between pairs of process communications at their points of contacts.

3.2.1. Types of gateways :

1. Some gateways simply read messages from one network (disembeding them from that network´s packaging and control formats), compute a routing function, and send messages into another network (embeding these messages following the control formats and packaging procedures of that network). Since the networks involved may be implemented using different media, such as leased lines

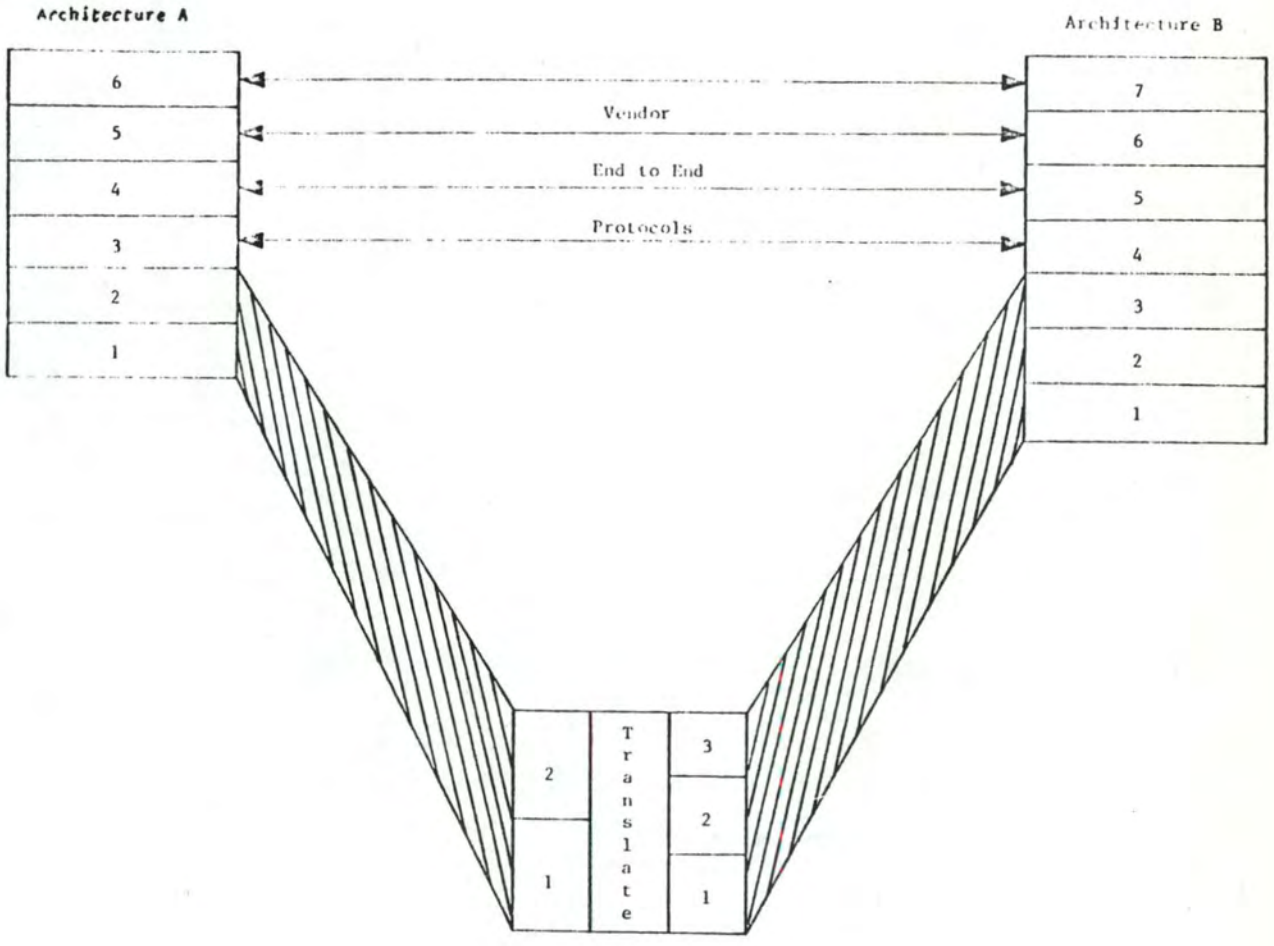


Figure 8. System Gateway Protocols

or radio transmissions, this type of gateway is called a Media-conversion gateway.

2. Other gateways may translate the protocols used in one network to the protocols used in another network by replacing commands received from one network with different commands, with the same protocol sementics, in the other network.
This type of gateway is called a protocol-translation gateway.

The distinction between media-conversion gateways and protocol-translation gateway is one of degree :

> the media-conversion gateways bridge the gap between differing links and physical level protocols, while protocol-translation gateways bridge the gap between differing networks and higher-level protocols.

3.2.2. Remark :

The translation approach to network interconnection is inversely correlated with the protocol level. The lower two levels, the physical, link and network levels, are hop-by-hop (node-to-node) in nature and present no interconnection issues in term of compatibility (a gateway has to support the first three levels protocols of each of the networks it interconnects).

1. To one school, the higher levels, Session, Presentation, and Application, have so many compatibility requirements that it seems quite unlikely that good interconnection of different protocols at those levels will be workable.
Thus it is at the network level and the transport level that the interconnection of network seem practical in regard to the diversity of the current network architectures. The network level provides for the packetizing messages, while the transport level handles end-to-end control. Thus the gateway solution really address only the transmission subsystem. This means the common carrier equipment would handle transmission and control of the packets without too much concern about their contents while the computers vendor's equipments would handle the data and data-flow contained in the packets.

2. But this view does not make possible the comprehension between end-users.

Another school wants to design Gateways at a higher level. It is to say, at a level where it is possible and meaningful to understand data information and control informations.

The gateway interfacing two or more networks (subnetworks), aims to realize relevent transformations for a protocol A to be transformed into a protocol B, semantically equivalent , inside another network.
The gateway receiving a message from network A, for exemple, will extract the meaningful data and control informations, interpret them, and transform them into an equivalent output message to network B.

The second goal is more complex than the first, but is the more powerful and interresting one. We can assume it is the long range objective to realize.

3.3. Validity of the layering decomposition

If there is an agreement on the use of layered structure as means of achieving connectivity, there is yet no consensus on the functional distribution of the layers. This results from the differing perceptions of market needs and the network solutions.

OSI Reference Model aims to unify those various views, but the OSI Reference Model has not yet reached a stable layering structure.

1. The network layer status has to be reviewed in order to cope with the different existing transport means.

2. The functions and structures of the upper layers (5,6,7) are controverted for, at less, three reasons.

3.3.1. Network layer

1. There is a common mistaken belief that X25 represents the lowest three layers specified in the OSI model.
   And that, for historical reasons: at the beginning the three lowest layers have been assumed to constitute an homogenous means materialized by X25. But ISO has had to realize that the transmission supports were not and would never be homogenous in the reality, and to consider the other existing transmission medias (circuit-switching, X21, ...) in reviewing its layers description, detaching it from X25 [ROSI] p77.

   The network layer is now assumed to contain functions necessary to mask the differences in the characteristics of different transmission and network technologies into a consistent network service. This service shall be the same at each end of the network-connection, even in the case of a network-connection spanning several subnetworks offering dissimilars services.
   Those recommendations concern the concatenation of networks but not the concurrent use of several networks by one system.

2. These multiple supports are meanwhile employed

concurrently by user systems (for instance a system connected to a PDN, a LAN and a private network). And OSI network layer has to take it in consideration.

Generally these medias are specified only for themselves and do not tell how to operate and manage a connection with another communication mean (for exemple, X25 describes its own protocols and procedures but don't explain how a system could use it concurrently with a circuit-switching network or an X21 connection).

For that reason, some people from ISO and ECMA think that the network layer should be broken into two sublayers achieving complementary goals.

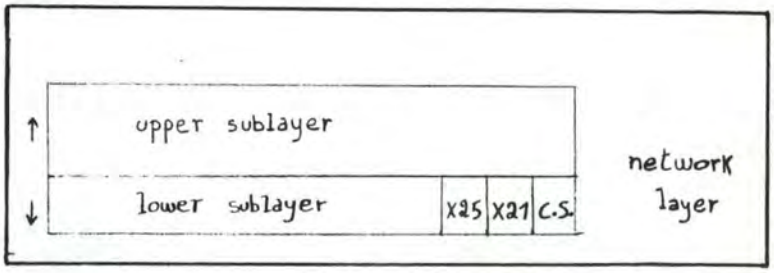


figure 9 : network layer decomposition

3.3.1.1. Lower part

The lower part of the layer would be downward oriented to handle with the various transmission technologies.
It would be a set of boxes adapted each to one transmission technology (packet-switching, circuit-switching, datagram services, leased lines, ...)

This sublayer aims

- to present an homogenous network perception to the upper layer, releaving it from the physical managing of those medias,

- to provide it with an easier virtual consistent network service,

- to manage the use of those transmission supports relieving the upper part of doing it.

3.3.1.2. Upper part

To solve the problem of the concurrent utilization of those medias, an additional protocol is needed. It would form the upper sublayer of the network layer. (OSI only speaks about connections that use several individual communication services in series.) This sublayer would provide the Transport layer with a well defined set of services that a Virtual network is due to support (connection establishment, data exchange, error control, ...). It will mask the concurrent management of the multiple medias - via the lower part-.

Its functions would be :

- interfacing with the lower part (processing requests from and/or responses to the upper layer)

- monitoring and managing the parallel and concurrent use of the available physical mediums : selecting one set or the other depending on its congestion, efficiency, reliability, service quality required, and/or other relevent parameters.

- managing the created virtual connections getting through the various medias.
  For example, a full-duplex connection using simultaneously an X25 connection an a datagram service to support the messages exchange. The layer should maintain sequencing, control, reassembling, ..., on both these medias, insuring the transport they operate as a pipe between the two end-systems.

But what is the best network service, the transport layer to provide with, is not clear; in particular for what concerns the routing and path management services.

Two schools propose differing solutions.

1. The first one supports the idea of a connection oriented service. It is similar to the permanent virtual circuit or virtual service of the packet switching networks, but is assumed here through a series of concateneted medias (a circuit-switching net followed by a packet switching net connected, through a LAN, to a private network for instance).

   It aims to trace a physical path, at connection time, from the sender to the receiver, in order to allow the communication. Once this latter is established, the path is maintened during the dialogue, assuming rerouting in case of physical failure.

   The managing of those pathes (setting up, maintaining, resetting,...) is quite heavy to handle, in counterpart of the easier addressing service provided.

   - For exemple, two systems intending to communicate. The network level of the sender selects an underlying circuit switching network to route the request call or connection call. This message flows through several subnetworks and finally reaches the receiver via a packet-switching port. If the communication is agreed, all the following messages will follow the same path. In case of physical failure of one part of the path (a leased line in an intermediary PDN or an entry node to a LAN) the network level has to reset the path and to reallocate a new route to help reestablish the dialogue.

2. The other school backs the Datagram service option.

   This technique is similar to the Datagram service in a Packet Switching Network.

   Self contained messages -with full address field- which can be fragmented into larger unit of signification, are transfered through the transmission medias from the sender to the receiver, without existence of a previous established communication.

   Each individual message can follow differents routes between source and destination. This gives each network and gateways the full flexibility to respond to congestion or failure by dynamically altering routing on a message by message basis.
   The sequencing is not performed and a function of messages reordering has to be done at reception. This function can be dedicated to the Transport layer (class 4 of services).

   This technique would simplify the network and routing control and the concurrent utilization of the

underlying medias, but it requires full addressing in each message and is not always adapted to large data flows. It seems also that the network is more sensible to congestion/saturation.

Another advantages is that in case of a route failure no session reestablishment is necessary, for each message follows its own route.

- For exemple, assume that a system is connected to a PSN, a PDN X.25, a PDN X.21, a CSN, and a LAN.

  Messages are to be sent to a remote system which can be reached via each one of the supports. The network protocol divides each message into segments with correct headers and then selects a relevent media to sent them. Sometimes it switches to another media in order to distribute the load. The fragments reach destination through different ways, after various delays. There the transport protocol reorders the fragments to reconstitute the message (1).

3.3.2. Transport layer

This layer definition is pretty well accepted by everybody.

The notion of a transport service is that of a universal communication interface which offers, to session layer, uniform facilities independent of the underlying communication mediums.

The Transport interface can be viewed as the upper interface of a transport protocol. The interface is constant, irrespective of the communication subnet services. It bridges the gap between services available and facilities desired. By defining a universal set of facilities the transport protocol provides thus a constant quality service to the upper layers and becomes a function of the underlaying service only (fig. 10).

To provide these standard services, the functions performed vary following the lower layer service quality

---

(1) : for more information about similar methods (advantages/disadvantages) consult " Issues in International Public data Networking" - Computer Networks 3 -1979 - p259;266 - Grossman/Hinchley/Sunshine - and "Computers Networks" - Tanenbaum - prentice hall - 1981 - p353;368 - p188;192

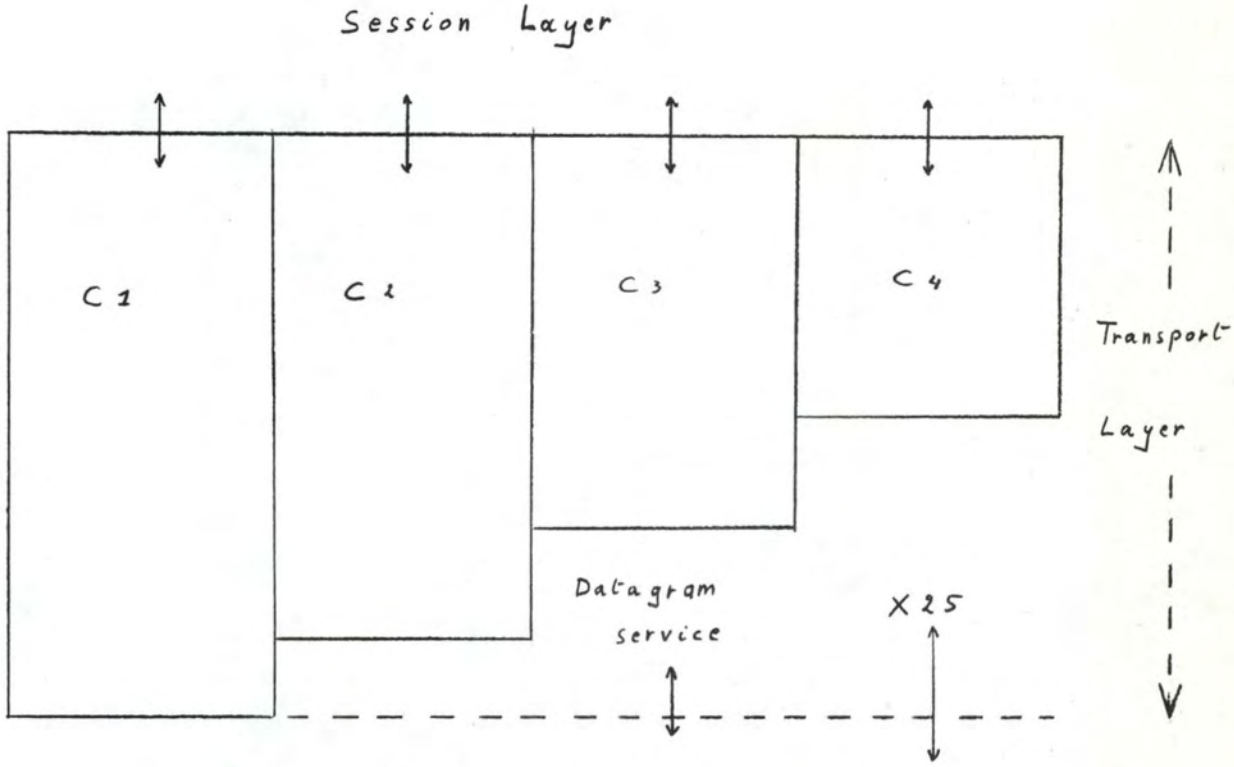


Figure 10 : classes of service depending on Network provided services.

relied on. Several sets of classes are so selectionable for that goal.

Meanwhile, a one service class Transport layer could be set up. This single class would provide a 'maximum service'. That means the functions would be implemented considering the lowest network quality service (no sequencing, no reordering). This would also fix the interface with the Network layer; the functions already performed in this latter would simply be bypassed in the Transport layer.

When the service quality required from transport layer is the same as the service provided by layer 3, the transport protocol becomes a zero functional level.

This layer provides transparent transfer of data between upper entities. It relieves the Transport users from any concern with the detailed way in which reliable and cost effective transfer of data is achieved.

3.3.3. Higher layers

The first four layers can be said independent of the processes, and applications they serve. This set of layers assumes a Transport function regardless of the signification of the data transmitted.

For what concerns the upper layers there is a goal change. This set of layers is upward oriented, aiming to serve the application, the end-user. The services, the protocols, the goals are thus, a function of both application and global criterias.

The purpose of these layers is :

- Application layer : "all the exchange of meaningful ... information".
- Presentation layer : "preserve meaning while resolving syntax differences".
- Session layer : "organize and synchronize ... dialogue and manage ... data exchange"[OSI RM].

3.3.3.1. Controversy

3.3.3.1.1. Layers structure

To the mind of some people the definition of those layers should be reviewed for at least three reasons that we will explain in the following.

1. Inseparability of application requirements from the functions provided by those layers.

2. Identified classes of applications require specific functions that can not be fully satisfied if approached in a global manner.

3. The vertical layering structure (Application layer, Presentation and Session layers)is not always adequate.

1. First point

If the objectives of the lower layers are well defined and totally independent of the informations transfered, and the decision criterias for the design well identified, it is not fully the case for the upper layers.

- For exemple, the dialogue management, in particular the 'turn of transfer' control mechanism (flip-flop). If we consider the dialogue between a remote Data Base and a user at a terminal, on a half duplex link: the changement of direction of sending is linked to the application itself.
  When the terminal has ended the sent of a query, the direction of transmission must change for the D.B. to reply. But this command depends upon the meaning of the data exchanged. How can the Session protocol know what the user asks is completed without knowledge of the signifiance of data received ? Here the criterions ruling the direction changement are part of the data context.

In fact, for those layers, designers do not have accurate criterions to specify which useful services to provide. Only the Application program, or programmer know what their requirements are.
The Presentation layer and Session layer have often to be piloted by the processing application. This is the case, for instance, for Data Bases updating (definition of restart points, ...). The existence of a Presentation layer can be, in some case, annoying: the application having to translate its own concepts to those understood by the Presentation protocol, and, at reception the

translation has to be done again. This is a source of possible misunderstanding between application levels.

2. Second point

Even assuming separability between application processes and Presentation and Session layers, and an independent designing of the two latters, it is doubtful that a global set of standardized functions, in each layer, could provide optimal services for several dissimilar classes of applications (file transfer, D.B. consultation, remote access, task to task communication, graphical applications, ...) at the same time.

It would be more suitable to identify specific classes of applications. And then to try to design corresponding protocols and services, application oriented (sometimes covering the three levels),and standardize them, plus a set of functions common to all classes.

3. third point

Assuming the effectiveness of the previous points; it is to say

- identification of global classes of applications and,
- more specific design of the protocol layers and service layers (enhancing capabilities);

some people underline that the structure order of the three upper layers is not always suitable.

They argue that in some cases, it would be more suitable to set the Session layer upon the Presentation layer, in order to better meet the reality ; and that, in other cases no layers at all are necessary to better suit the Application requirements.

To make that clear, imagine the communication, through a network, between an asynchronous terminal and a remote host.
This terminal has a special code set (EBCDIC, ...) different from the standard codes of the network and the host. The relevent code translation is achieved by the presentation layer at one end (usually the local

terminal). (The host must receive the commands of the terminal in the code it recognizes -the semantic of the command being respected- and reverse)
But some commands, a break or control characters, have to be handled by both the terminal's application and the network providing the connection. For instance if the break means a reversing of the transmission right (flip-flop discipline), the Session level has to know it. Those commands should, then, be recognized at Session level sometimes.
But before to be interpreted by the Session Protocol entity, they may have to transit through the Presentation level to be understandable. It is not always suitable, it could cause a synchronization loss; the processing time of the command being too long for continuing to correctly handle incoming data flow.

For some applications, the Session level should then be upon the Presentation level.

If OSI grows following those principles, the structure of the upper layers would be modified to be the following (fig. 10b):

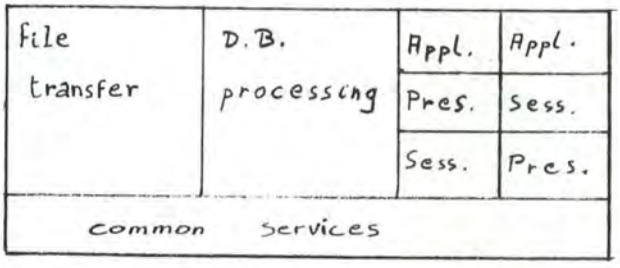


fig. 10b : alternative layering structure

3.3.3.1.2. Concluding remarks :

- a reply to those willing, for the keeping of the present structure, is that having a standard implies the acceptation of some restrictions in order to realize Systems Compatibility.

- Perhaps one mistake in the OSI field of the upper layers is to aim to define standards, services and protocols having to compromise with the universe of existing application types, which is not always necessary nor easy. It would thus be suitable and more realistic to identify large Application profiles and standardize them ,and after, review the Layer specifications to better meet the requirements identified.

- The controversy is not ended; it is to remark that in a recent paper (1) the ECMA tries to address the issue "what is the nature of the upper layers of OSI ?".

  This article offers a clarification and tends to reconcile the apparently diverging views.
  It recalls that the objective of ISO SC16 is "to standardize the relevent protocols between systems" and not to standardize the associated internal structure of systems.

  If the OSI Reference Model describes externally visible layers of protocols, it does not require them to be implemented following a layered concept. (" ... the necessary separation of different aspects of OSI into separate specifications, and thereby into separate Protocol Control Information formats, neither requires nor even necessarely implies separate implementation mechanism for each layer.")

  The forgetting of that concept is an ingredient of the misunderstanding about upper layers.

  They are opposed to "so-called optimizations which delete Session and Presentation layer structure content", and underline that fact with an exemple related to F.T.P. (File Transfer Protocol).

  In few words,

  1) they agree with the fact that specific applications exist and require each some particular needs in each of the layers.

  2) they underline that parameters, formats, subset of services common to those various

---

(1) - ECMA / TC23 / 81 / 17

applications, and/or independent from them, exist.

3) Both those subsets have to be integrated in the layer they are related with (Presentation layer or Session layer), in order :

a) to assume a valuable and understandable modeling description, and
b) to preserve the separateness of each layer of protocol and that of the related specifications.

4) Any application should use those common techniques which are defined in the layering architecture, and not invent its own encodings, dialogue structures, etc ...

5) The individual implementor makes his own choice whether to integrate all the associated software into a unit customised and optimized for its application(s), or else to use separate more generalpurpose units of software with defined internal modularity and interfaces, etc ...

At the light of this article, the two first points of the previous argumentation seem to be no more relevants.

The third, in the other hand, can not be so easily disproved. WE could say that it is up to the implementor to cope with those problems, independently of the OSI Reference Model description. But the controversy is always opened.

4. CONCLUSIONS

OSI Reference Model has to forge ahead to reach a steadier state of completion.

But will it grow to a full set of standards, will it address gateways or will it sink into oblivion, only the future will tell it.

One point sure is that lot of people (manufacturers, national bodies, PTT's, ...) back its grow and act for that goal. Evenmore the Reference Model has a strong inner consistency. And this quality seems to be sufficent to ensure its future developments.

That the existing architectures be modified to match the OSI standard seems unprobalble. It is more presumable that they will provide a gateway or another mechanism matching the OSI requirements. The existing systems would thus remain incompatible in their own, but would ensure compatibility through the use of system/OSI gateways.

For what concerns the new created and future architectures, especially within the local network area, more likely will they match directly the OSI standard in order to assume heterogenous compatibility for a larger disponibility.

Chapter 3: ARCHITECTURE DESCRIPTIONS

1. INTRODUCTION

Four architectures will be developed in the following. These are SNA, DNA, DSA, CNA.

In a first section, the structuring principles used for those separated descriptions will be brievly exposed.

The four following sections will be devoted to the respective architectures. And in the last part cross comparisons of those various architectures will be addressed.

We point out that the separate descriptions are not included herein, but in the Appendix A of this dissertation. This is done in order not to overcome the limited amount of pages granted for printing.

- Remark :

  Considering the various sources of information concerning those architectures, the same description degree cannot be reached in what concerns the details.

  This is due, for part, to the reluctentness of some constructors to release technical information, or to the the delays needed to obtain this latter, and for other part to the lake of time needed to cover and complete these descriptions.

  Evenmore, the informations at disposal are not always up to date. We have thus to appologize for that making some points no more relevent.

  The descriptions of CNA and SNA are not complete for the reason specified above.

2. STRUCTURING PRINCIPLES

The principles used for the descriptions hereafter have been derived from [TUC SP]. This section constitutes a short overview of these latters. For a complete approach, refer to [TUC SP].

The systematic of these Structuring Principles is based on the notion of cut. A cut separates an inworld from an outworld and defines discrete interaction points, at which the inworld and the outworld may interact with each other [TUC SP].

Cuts serve for:

- identifying the units able to interact with each other and their interaction points,
- relating the interaction points of interacting units, and
- defining meaningful interactions at these interaction points.

The cut is the mean to completely abstract from the internal structure and details of the inworld of these units, and enforces to describe the meanings of interactions between these units in terms of external visible behaviour of these units at their interaction points.

The cuts we will use are the System cuts, the Service cuts, and the Protocol cuts.

1. System cuts :

   System cuts serve for achieving a topological decomposition of the real world, or to create mappings between the communication architecture and the real world. They identify individual systems (End systems and Transit systems) as being representatives of those physical components of the real world hosting individual pairwise communication activities (central units, switching devices, transmission lines, ...).

   End systems are systems hosting the communicating Application Entities, i.e. the representatives of communicating parts of the real world.

   Transit systems play the role of being the common media for transmitting these informations between End systems; they perform transmission and switching activities required for the information exchange between communicating End systems.

2. Service cuts :

Service cuts serve for achieving a functional decomposition of individual pairwise communication activities; they identify functional layers.

3. Protocol cuts :

Protocol cuts serve to coordinate system cuts and service cuts with respect to individual pairwise communication activities; they identify protocol entities.

The purpose of Service and Protocol cuts is to define a virtual structure for systems, thus determining their communication behaviour. The purpose of this virtual structure is to describe the structure of communications they must be able to maintain.

3. SNA DESCRIPTION

3.1. Introduction to SNA

SNA defines a unified set of commands, procedures, message formats and protocols used to facilitate data communication between SNA compatible products [IBM C].

SNA [Cypser 78], is a network architecture intended to allow IBM customers to construct their own private networks, both hosts and subnet.

Prior to SNA, IBM had several hundred communication products, using three dozen teleprocessing access methods, with more than a dozen data link protocols alone. The basic idea behind SNA was to eliminate this chaos and to provide a coherent framework for loosely coupled distributed processing.

Given the desire of many IBM´s customers to maintain compatibility with all these (mutually incompatible) programs and protocols, the SNA architecture is more complicated in places than it might have been had these constraints not been present.

SNA also performs a large number of functions not found in other networks, which, although valuable for certain applications, tend to add to the overall complexity of the architecture.

SNA has evolved considerably over the years, and is still evolving.

A SNA network consists of a collection of machines called Nodes, of which there are four types approximately characterized as follows:

- Type 1 nodes are terminals.
- Type 2 nodes are controllers, machines that supervise the behaviour of terminals and other peripherals.
- Type 4 nodes are Front End processors, devices relieving the main CPU of communication work.
- Type 5 nodes are the main hosts themselves (some controllers have some host-like properties in reason of distributed processing).

Each node contains one or more Network Addressable Units (NAUs). A NAU is a piece of software that allows a process to use the network. It is an entry point into the network for user processes.

SNA appears to be a centralised controlled, hierarchical network.

We can identify four physical components :

1. Host nodes :

   Equivalent to a CPU with Operating System, Access Methods,and a Data Base. It is responsible for Data processing, D.B. processing and communication system network management.

2. Communication Controller Nodes (CUCN) :

   Responsible for many Communication System (CS) functions as control of the remote network, that is attached to it; acting as a slave of the host node for which it carries out instructions and messages.

   it is also responsible for

   - control of the communication lines, deleting and inserting characters.
   - code translation.
   - activation/deactivation of lines.
   - error recovery.

   It can act as a F.E.P. to the H.N. (Host Node).

3. Cluster controller Nodes (CCN) :

   - provide remote locations with access to the D.B. or services at the H.N.

- consist of programmed controllers supporting devices and containing data and processing storage.
- process data and act as stand alone systems servicing their terminals.
- linked to the CUCN by SDLC lines.
- exemple : a 3600 firmware communication system

4. Terminal Nodes (TN) :

- send and receive data from/to the H.N.
- support the attachement of some devices.
- linked to the CUCN by SDLC lines.
- not programmed.

End Systems are H.N., CCN and TN .

Transit systems are CUCNs.

3.2. System cuts, Service cuts :

For what concerns those sections refer to Appendix A.

4. DNA DESCRIPTION

4.1. Introduction to DNA

DECnet [Wecker,1980] is a set of programs, protocols and hardware produced by Digital Equipment Corporation. The architecture of DECnet is called DNA (Digital Network Architecture).

The intention of DECnet is to allow any DEC's customers to set up a private network. A DECnet is a collection of machines (called nodes), with their O.S. and software modules, some of which may run users programs, some of which may do packet switching or batch. The functions performed by any given machine may even change in time.

DNA consists of a model, a set of interfaces, and a set of protocols. The DNA model describes a structure that embraces the software modules which perform networking functions for each DEC O.S. . The structure is layered and conforms for a major part to the OSI architecture.

DEC network links computers running different (but compatible) Operating Systems. The figure 25 showns a six nodes meshed network.

The DECnet implementation at each node acts as an interface between the node's O.S. and the network (fig. 26) converting DECnet formats to those recognizable by node's O.S., and reverse.

DNA appears thus to be a nodes' distributed network, topologically hostless. In terms of System cuts those identified are :

4.1.1. Nodes :

Nodes are DEC computers running their own O.S. which

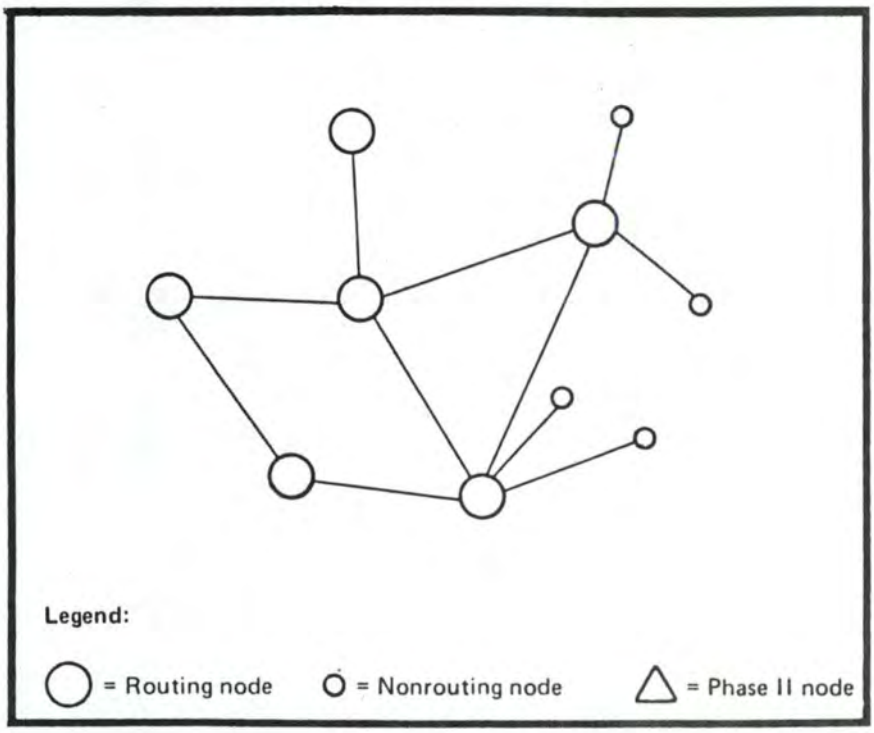


Figure 25 : A 6 Nodes Configuration

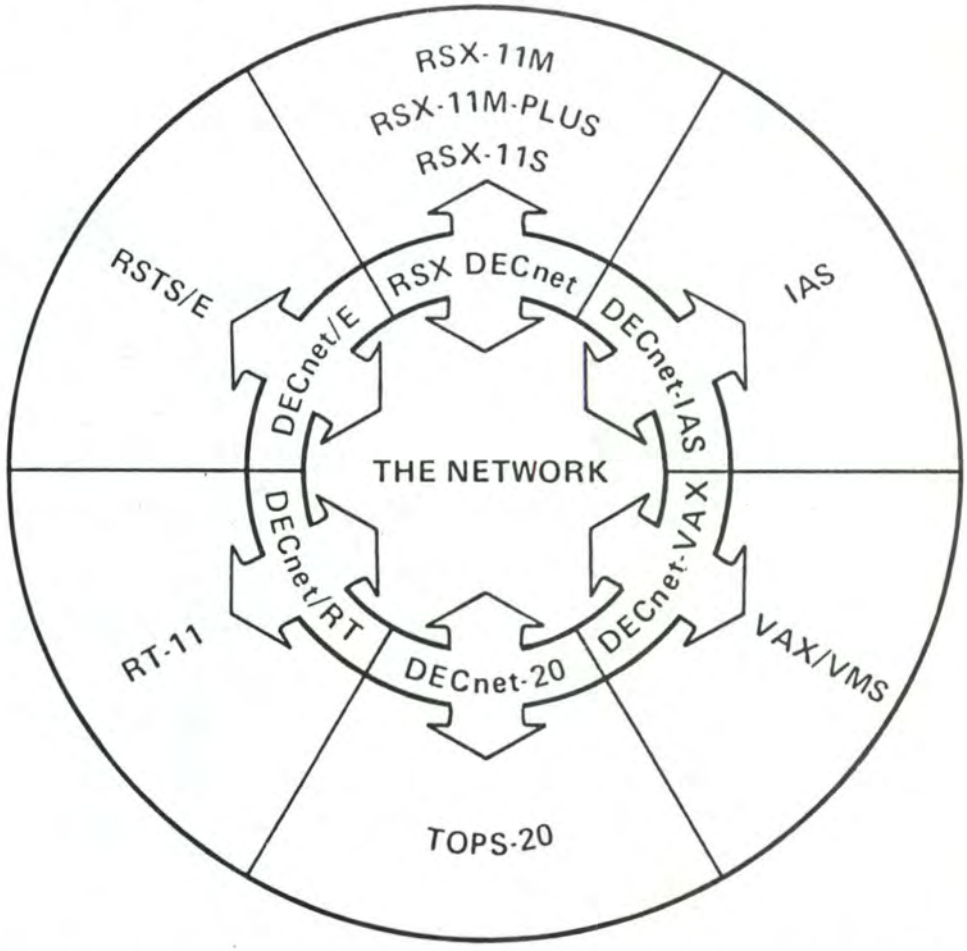


Figure 26 : DecNet Interaction with various O.S.

allow interactions through DECnet.

DEC doesn't identify nodes devoted to data processing and others devoted to transport functions. A node assumes the two functions, acting as a Data processor for local applications (80% of resources) and as network processor assuming networking functions (20% of resources).

In this context three subtypes of nodes are distinguished :

1. Routing nodes : a routing node can forward packets to other nodes in the network and can be adjacent to all other types of nodes.

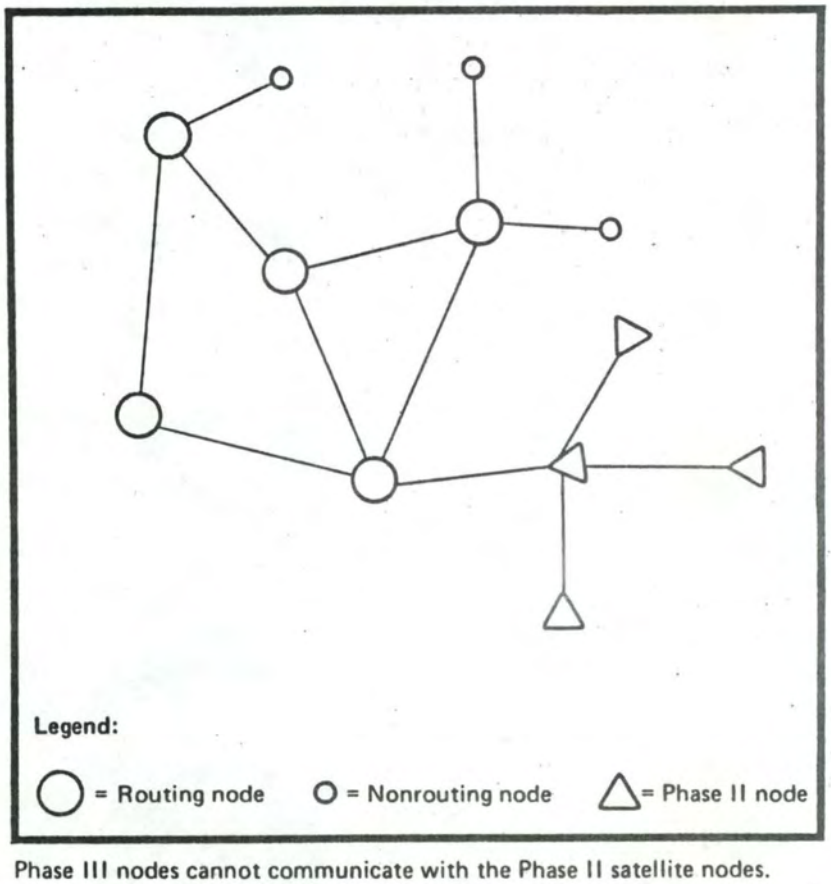
2. Nonrouting nodes : a nonrouting node can sends packets to other nodes in the network but packets can not be forwarded or routed through it. It can be adjacent to one other node only, and is therefor an end-system in a configuration.

3. Phase-II node : runs a previous phase-II implementation of DECnet and therefore does not support full routing. It can send packets only to adjacent nodes and cannot forward packets it receives onto other non-adjacent nodes it the network. It can be adjacent to one or more full routing nodes and/or to other phase-II nodes. Logically it is an end-system node in a Phase-III configuration (fig. 27).

- Remarks on network philosophy :

As above mentioned DNA is totally distributed . That is, there is no inherent central control functions (i.e. SSCP in SNA) in the network. To achieve this topological independence, the control and maintenance functions are executed at the level of usr's applications within the DNA structure (Network Management Layer).

A second characteristic is that all nodes are addressed uniformly. The network has no inherent



Phase III nodes cannot communicate with the Phase II satellite nodes.

Figure 27 : A Mixed Configuration: a Phase III Network Adjacent to a Phase II Star-shaped Network

notions of a backbone communication network (i.e. no PU types as in SNA). The notions of host nodes, concentrator, and communication switching nodes are logical ones and depend on the software, as said before. Two nodes can change from host-host relationship to a host-FEP relationship without affecting the user or network software. The transport level communication protocol and addressing are the same for both these situations.

These characteristics are achieved by having DNA built around the following principle : all network usage can be modeled as communication between application level processes. These application level processes are called resource objects and may be application programs (tasks), operator or I/O devices. To this end, there are several ways in which computers in a network can work together (fig. 28).

- In the program-to-program mode , a program in one node or computer requests a program in another to perform a data processing task, and the result are retained. That often involves gaining access to a data file.

- In the file-transfer mode, the first computer retrieves a data files from a storage device located at the second computer and computes locally. The same computing task can be performed in either mode.

- The third mode in which computers can work together, resource access, describes their ability to share network resources files, line printers, terminals, graphic plotters, and application programs. Typically, resources access permits one computer to retrieve a single record from a disk file linked to another computer as if the file belonged to the first computer. The first computer may requests that a line printer, controlled by the second computer, print out a report. If instead, the first computer wants the second to create a report file and return that file in the file transfer mode to the local site for printing, it can request this in the program-to-program mode..

No matter what type of traffic flow (interactive, real-time, or batch is occuring, the flow is always in one of these three modes.

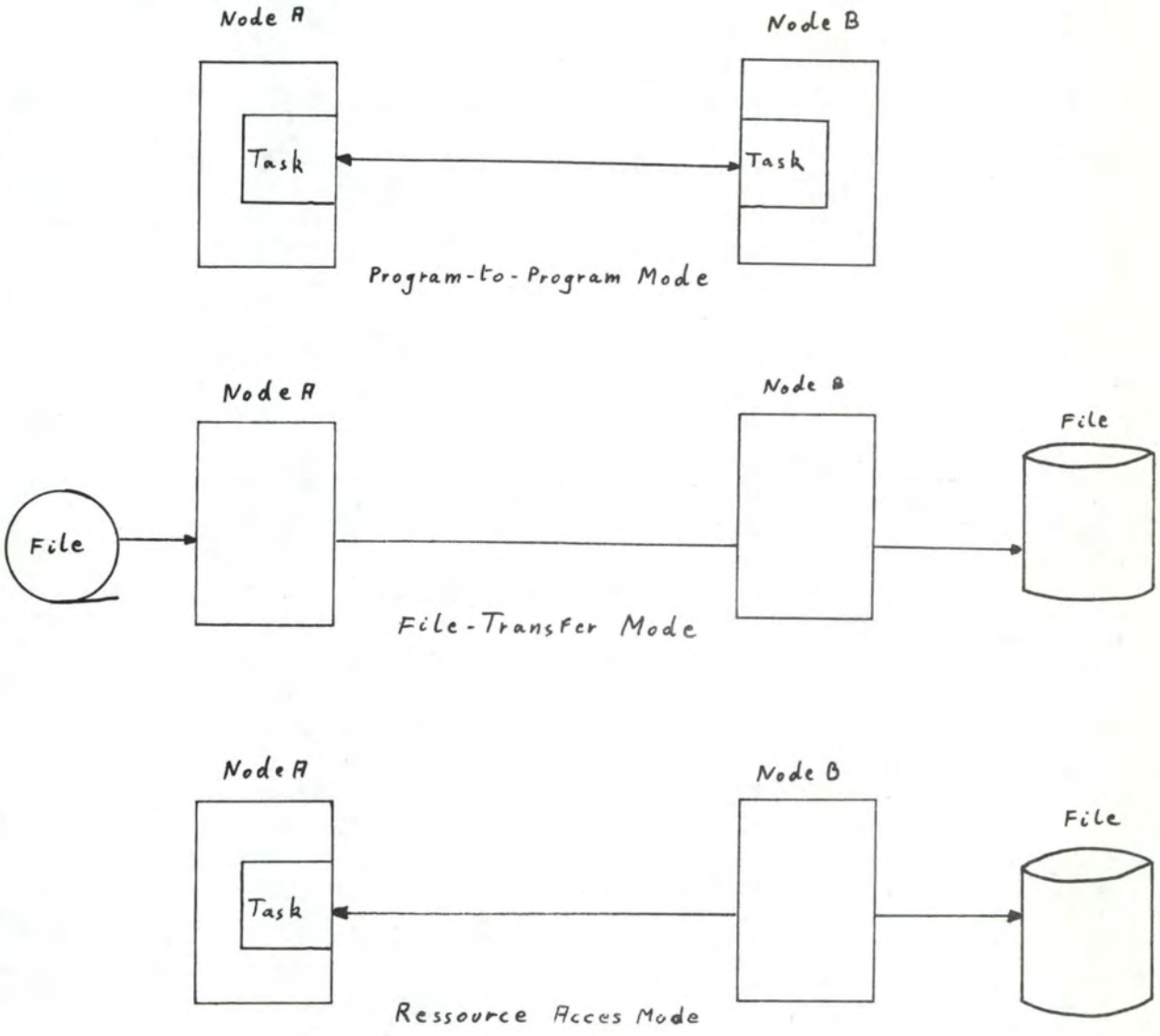


Figure 28. Resource Object Modes

- Configuration :

A DNA network [DNA DTP] consists of two or more DEC processor nodes each loaded with DECnet software product compatible with its operating system. Each DNA interprocessor operation utilizes a layered architecture and a common set of DECnet protocols. The range of functions that can be performed between any two nodes is limited to the functions they share (the network as a whole, however, is not limited to the functions common to all).

4.2. System cuts, Service cuts, Protocol cuts :

For what concerns those sections, refer to Appendix A.

5. DSA DESCRIPTION

5.1. Remark :

As said above in preliminary introduction, no sufficient information have been made available from CII-HB. Due partially to the fact that existing DSA descriptions are being revised.

5.2. Introduction to DSA

Distributed System Environment (DSE) is CII Honeywell's master plan for distributed processing. It is a set of protocols, programs, and hardware used to create an environment including information proccessing, data management (file creation, manipulation, storage, movment in DSE) and network proccessing (network administration and management).

DSE allows customers to set up a network, linking their systems, in order to insure communications between them. DSE supports the integration of a variety of systems and communications Protocols into a single, distributed network which supports interactive, batch or time-sharing applications simultaneously.

The architecture of DSE is called Distributed System Architecture (DSA). DSA provides a universal set of rules that govern the data movement within a DSA network.

The basic structure of DSA conforms to the International Standard Organization (ISO) OSI Reference Model, to which Honeywell has expressed a firm commitment. Using this model, DSA defines the functions performed by each layer, the protocols which control the dialogue within each layer, and the interfaces between the layers.

5.3. System Cuts

The DSA network consists of nodes (computers and terminals) joined by links, as the other existing networks.

Communications between nodes can be via private or public communication facilities. DSA provides specific interfacing for dedicated lines, for public X21 circuit-switching networks, public packet-switching networks such as Transpac or DATEX-P. Both Datagram type and X25 packet-switching are supported, with DSA node acting as a Gateway.

The nodes can be minicomputers, network processors or terminal controllers.

Each node has a unique identifier within the network, and within a system (or node) activities (programs, terminal users, operators, ...) have local identifiers.

The network is divided in two parts by CII-HB: (fig. 51)

1. the Primary Network which interconnects the nodes communicating by use of DSA rules.

2. the Secondary Networks which contain terminals and/or terminal clusters connected to a node (primary node) or between them via switched or dedicated lines or even via packet-switching networks.

In terms of System Cuts, we can identify the followings: the hosts, the network processors, the satellites, and the terminals.

5.3.1. Host :

A host is data processing oriented; it is configured with a channel-attached Front-End (FE) or integrated communication processor.

It provides services both to local and terminal users,

F51
F52

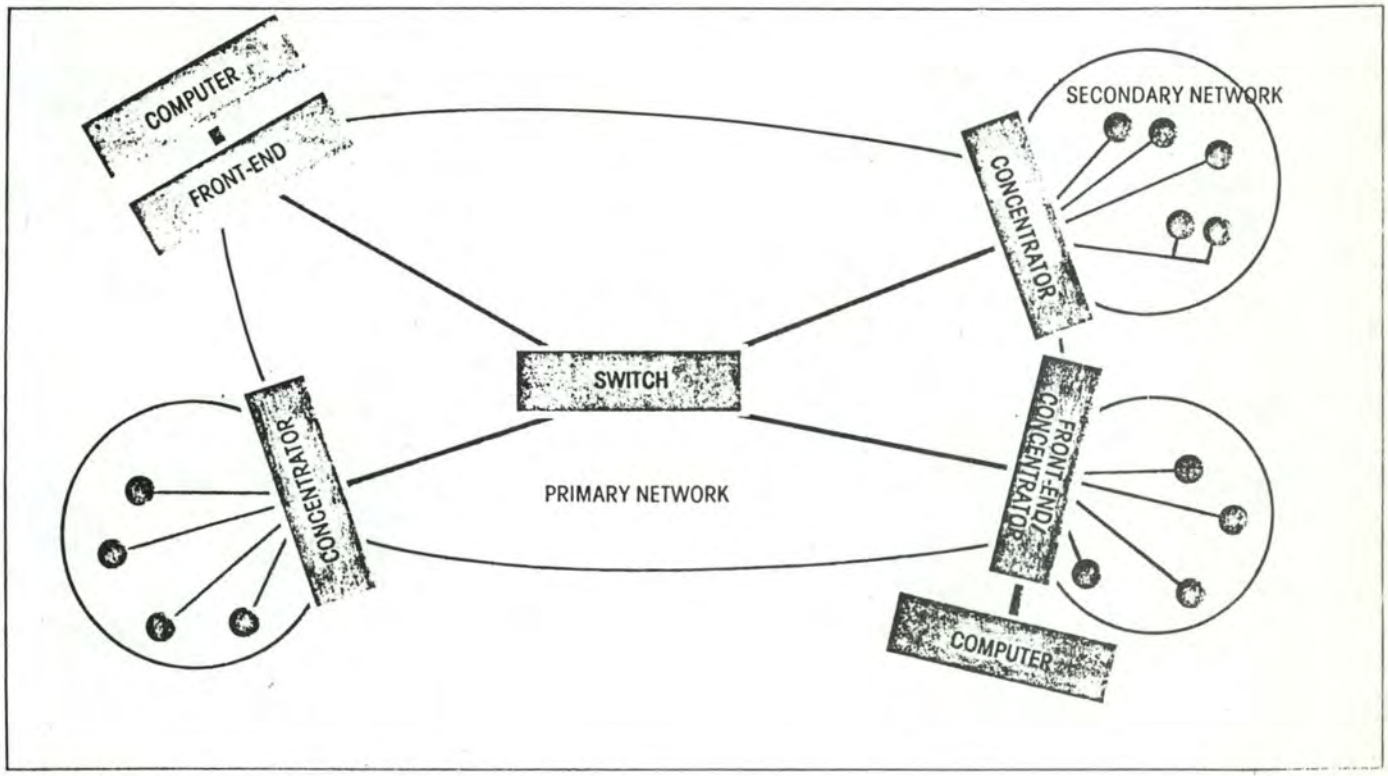


Figure 51. Network processor roles.

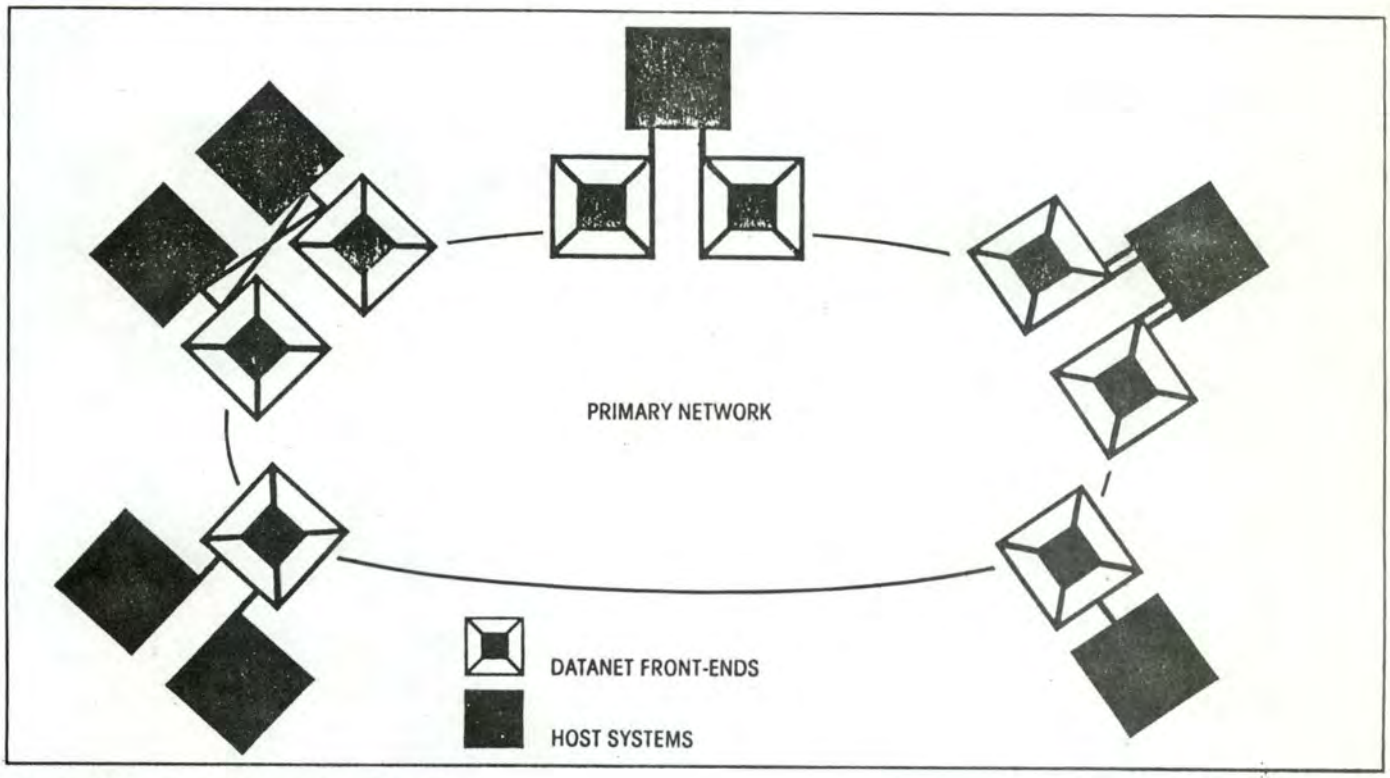


Figure 52. Front-end processor configurations.

and to other hosts or nodes in the network (the services accessed through the network transit via the FE).

CII-HB considers a host to be a large, medium or small scale computer with storage capacity. These could be DPS 8, DPS 7, 64 DPS systems of the CII-HB products.

Generally the host implements the application and message management layers (see forward sub-section 3) exepts for the pre-DSA systems (like DPS 8).

5.3.2. Network Processors :

These are computers dedicated to communication management functions. They can perform any combination of the three following roles :

1. Front-End processing providing network services to the hosts, to which they are attached, allowing them to concentrate on data processing.
   They also serve as interface between one or more hosts and a primary or secondary network (fig. 52).

   - remark : acting as a gateway for non DSA host, a network processor (FE) contains both the communication management layers and a specialized data management software interfacing betwenn the DSA and non DSA elements.

2. concentration: for acces to DSA primary network by terminals and possibly computers, using non DSA communication techniques,located in the secondary network.

3. switching : providing routing services and network management services in the primary network.

The relations between the hosts and network processors can be 1-1, 1-N, N-1, M-N.

5.3.3. Satellites :

A satellite system serves a number of local users (terminals, ...) in its secondary network, and communicates with hosts and other satellites via the primary network, obeying DSA rules.

It provides its local users with processing, storage facilities, access to resources, within remote systems (hosts). It support a wide range of networking applications such as file transfer, remote job entry, ... , in addition to the users defined applications. A satellite does not support data processing for non local users.

5.3.4. Terminals :

A terminal is a device directly connected to a host or a satellite, or virtually connected via either a network processor, a terminal controller, a satellite to any host in the network.

It provides its users with access to programs and/or services in any processor of the network.

5.4. Service cuts, Protocol cuts :

For what concerns those sections, refer to Appendix A.

6. CNA DESCRIPTION

6.1. Remark :

No sufficent information has been made available in time. This is due to the fact that CNA concepts and architecture were being revised.

6.2. Introduction to CNA

Communication Network Architecture (CNA) is the network architecture developed by NCR. "CNA is a direct result of strategy (business plan). CNA is a technical plan that ensures the connectability of products by defining the communication interrelationships among NCR products. CNA reflects the corporate business objectives in the specification of these communication interrelationships. The scope of CNA includes all those functions and equipment which enable people, processes and devices to communicate".

CNA is defined by a set of formats, protocols, programs and hardware used for networking in various environments (SNA, Packet-Switching, Circuit-Switching, BSC, Start/stop, OSI and DCNA environments).

CNA is presented as a "conceptual template, or meta-architecture, used for the primary purpose of describing how multiple, widely differing communications architectures and technologies fit together and intercommunicate under CNA". "This meta-architecture provides a structure which defines the functional and topological bound of CNA, but also allows different environment expansions".

The structuring of CNA architecture model seems to be similar to the OSI one, even if this latter is described as a particular environment to cope with.

It is to point out that CNA includes gateway services, called Service groups, which provide a means, a conceptual mold into which the formats and protocols of disimilars environments (systems) can be mapped.

The layered structure of CNA, its major services, and its relation to OSI layered architecture are shown in figures 57 and

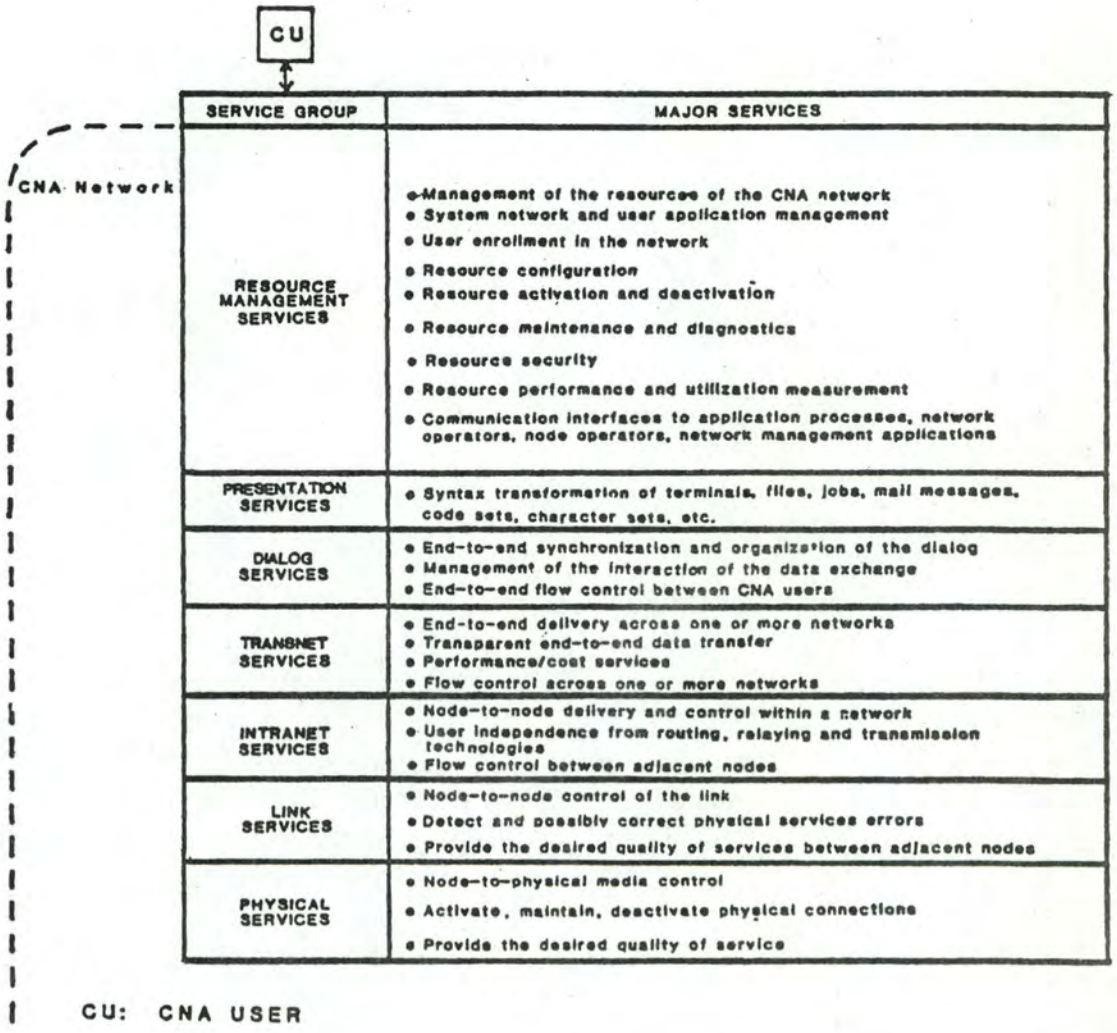


Figure 57. Summary of CNA Services

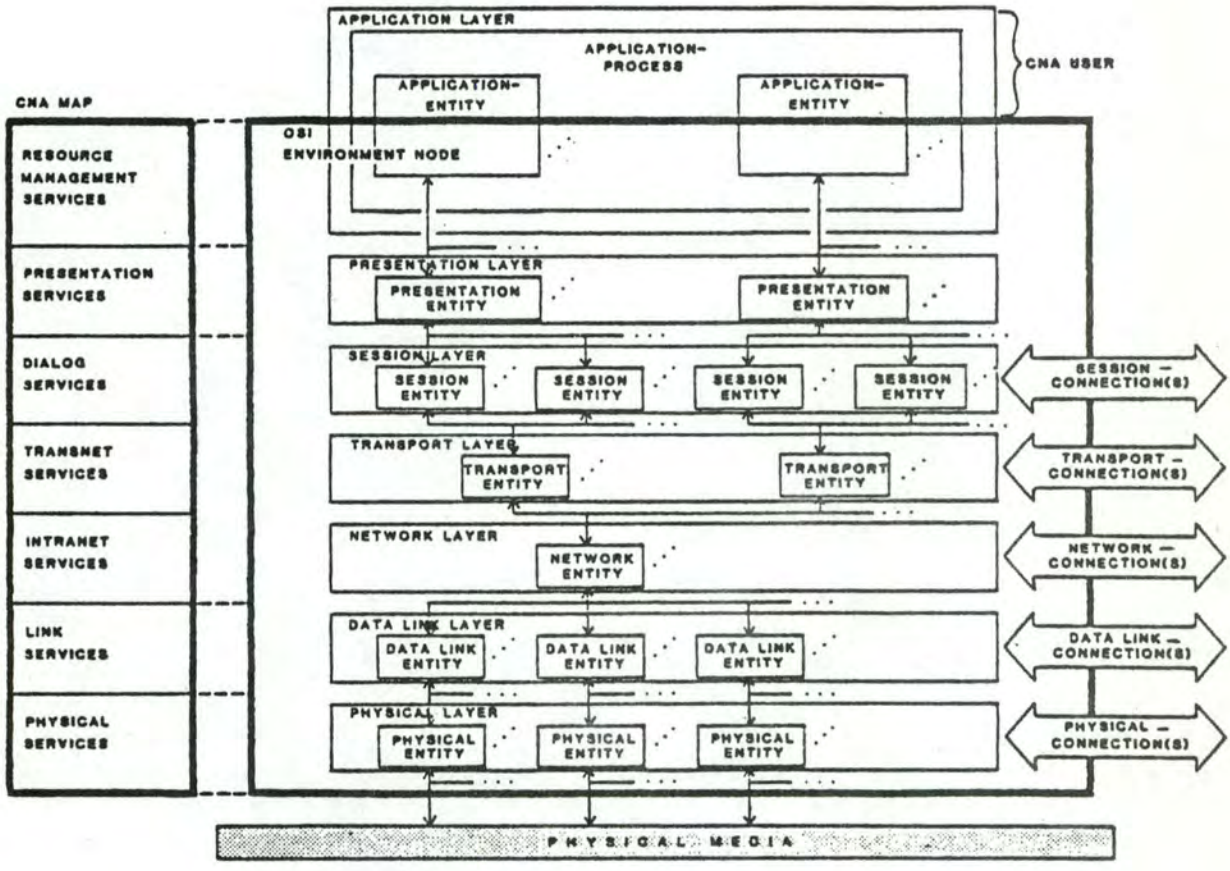


Figure 58. Overview of an OSI Environment Node and Its Relationship to CNA

58 respectively.

6.3. Service cuts, System cuts, Protocol cuts

For what concerns the Service description of those layers, refer to Appendix A. No information relating to Protocols is available.

7. ARCHITECTURAL COMPARISONS

7.1. Global approach

	DNA	SNA	DSA	CNA
NETWORK LEVELS				
One level network	X	X		
Two level network			X	
NETWORK CLASS				
Private network	X	X	X	X
Marketed network	X	X	X	X
Public network				
NETWORK TOPOLOGY				
Star topology		X		X
Ring topology				
Complete meshed topology	X		X	X
Partially meshed toology	X		X	X
Hierachical topology	X	X	X	X
TRANSPORT TECHNOLOGY				
Circuit-switching			X	X
Message-switching				
Packet-switching	X	X	X	X
ROUTING				
Floodind routing				?
Fixed routing		X		?
Random routing				?
Isolated routing	X			?
Distributed adaptative r.	X		X	?
Centralised adaptative r.		X		?
Delta routing				?
Hierachical logical routing				?
Hierachical physical routing				?

	DNA	SNA	DSA	CNA
NETWORK TYPE				
Terminal network		X	X	X
Computer network	X	X	X	X
COMPATIBYLITY				
Homogenous	X	X	X	X
Heterogenous				X
DATA INTEGRITY & CONFIDENTIALITY				
TRANSMISSION INTEGRITY				
Point-to-point encryption	2	X	2	2
Throughout network encr.	2	X	2	2
End-to-End encryption	2	X	2	2
AUTHENTICATION				
User		X		X
for password communication		X		2
for encryption communic.				2
ACCESS CONFIDENTIALITY				
for password	2	X	2	2
for encryption	2		2	2

2 : unknown

	DNA	SNA	DSA	CNA
LAYER 2 FLOW CONTROL				
RESOURCES MANAGEMENT				
Statical		X		2
Dynamical	X		X	2
DATA FLOW MECHANISM				
Send & wait				2
Credit	X	X	X	2
Rate				2
Class				2
LAYER 3 FLOW CONTROL				
RESOURCES MANAGEMENT				
Statical				2
Dynamical	X	X	X	2
DATA FLOW MECHANISM				
Send & wait				2
Credit	X	X	X	2
Rate				2
Class				2

2 : unknown

	DNA	SNA	DSA	CNA
LAYER 4 FLOW CONTROL				
RESOURCES MANAGEMENT				
Statical				2
Dynamical	X	X	X	2
DATA FLOW MECHANISM				
Send & wait				2
Credit	X	X	X	2
Rate				2
Class	X	X	X	2
LAYER 5 FLOW CONTROL				
RESOURCES MANAGEMENT				
Statical				2
Dynamical	X	X	X	2
DATA FLOW MECHANISM				
Send & wait				2
Credit		X		2
Rate				2
Class	X	X	X	2
NETWORK DISPONIBILITY				
Experimental network				
Constructor marketed netw.	X	X	X	X

2 : unknown

7.2. Services Provided in reference to the OSI Reference Model

The comparisons will be dived into with the help of tables, proceeding layer by layer.

7.2.1. PHYSICAL LAYER

The services provided by the various architectures are similars.

1. IBM SNA : interfaces X21 and X21 bis or V24
2. CII-HB DSA : those corresponding to CCITT recommendations
3. DEC DNA : those corresponding to CCITT recomendations

7.2.2. DATA LINK LAYER

1. SNA : achieves the service of OSI Reference Model Data Link Layer restricted to Unbalanced Mode / Normal Response Mode
2. DSA : assures HDLC Lap-B (Balanced Mode) Protocol and services.
3. DNA : does not implements a bit based transmission Protocol but a Byte based one (HDLC like) called DDCMP.

7.2.3. NETWORK LAYER

OSI Services & functions	DNA	SNA	DSA	CNA
Network address	TL	PC	NL	IS
address translation		PC		?
network connection	Connec-tionless	PC	NL	IS
network-connection-endpoint-identifier	TL	PC	NL	IS
data-unit-transfer	TL	PC (PIU)	NL	IS
quality of services parameters	TL	1	NL	2
error notification	TL	1	NL	2
sequencing	/	1	NL	2
flow control	TL	1	NL	IS
expedited data units	?	1	NL	2
reset	?	?	NL	2
termination services	0	DFC	NL	IS
routing & switching	TL	PC	NL	IS
upward multiplexing	TL	PC	NL	2
downward multiplexing		PC		2
segmenting & blocking	TL	PC	NL	2
error detection	TL	PC	NL	2
error recovery	/	PC	NL	2

TL : Transport Layer - PC : Path Control - NL : Network Control
IS : Intranet Services

0 : connectionless
2 : depends on environment faced. Look at CNA description.

7.2.4. TRANSPORT LAYER

OSI Services & Functions	DNA	SNA	DSA	CNA
Transport Connection Establishment & Termination	SCL/NSL	PC	TL	TS
mapping of Transp. add. onto Network address	SCL	PC	TL	TS
Upward multiplexing	NSL	PC	TL	TS
Class of service Selection	/	PC	TL	TS
TSDU transfer	NSL	PC	TL	TS
Quality of service & error notification	/	PC	TL	TS
Expedited TSDU transfer	NSL	?	?	?
End-to-End sequencing	NSL	PC/TC	TL	TS
Flow control	NSL	PC	TL	TS
Delivery notification	?	?	?	?
Error detection & recovery	NSL	PC	TL	TS
Segmentation & Blocking	NSL	PC	TL	TS
Purge	?	?	?	?
Addressing	TL	PC/TC	TL	TS
Identification (connection)	SCL		TL	?

SCL : Session Control Layer - NSL : Network Service Layer
PC : Path Control Layer - TC : Transmission Control
TS : Transnet Services

7.2.5. SESSION LAYER

OSI Services & Functions	DNA	SNA	DSA	CNA
Session connection establishment/termination	SCL	TC	SC	DS
Mapping of session connection onto network connection	SCL	NAU SL	SC	DS
Normal Data Exchange	SCL/NSL	TC	SC	DS
Expedited Data exchange	SCL/NSL	DFC/TC	SC	?
Dialog management (TWS,TWA,OW)	NSL TWS	DFC	SC	DS
Context/interaction management	SCL/NSL	DFC	SC	DS
Quarantine service	/	DFC	SC	2
Session connection synchronization (reset, ...)	/	DFC	SC	DS
Exception reporting		TC	SC	2
Upward multiplexing	NSL		SC	2
Data delivery confirmation		DFC	SC	?
Sequencing	NSL	DFC/CPM	SC	2
Blocking		DFC	SC	2
Error detection & recovery	NSL	CPM	SC	2
Security	?	DFC	?	?
Flow control				DS

SCL : Session Control Layer - NSL : Network Service Layer
TC : Transmission Control - NAU SL : NAU Service Layer
DFC : Data Flow Control - CPM : Connection Point Manager
SC : Session Control - DS Dialog Services

2 : supposed to provide something similar, but no relevent information is available.

7.2.6. PRESENTATION LAYER

OSI Services & Functions	DNA	SNA	DSA	CNA
Presentation connection	NAL		PL	PS
Presentation Image negotiation	NAL	NAU SL	PL	PS
Renegotiation of image		NAU SL	PL	PS
data tranformation	NAL	NAU SL	PL	PS
data formatting	NAL	NAU SL	PL	PS
Syntax selection	?	NAU SL	PL	PS
Command formatting	?	NAU SL	PL	PS
compression	/	NAU SL	PL	PS
encryption	/	NAU SL	?	PS
Virtual Terminal Services	NAL	NAU SL	PL	2
Virtual File Services	NAL	NAU SL	PL	2
Job transfer & manipulation Services	NAL	NAU SL	PL	2

NAL : Network Application Layer - NAU SL : NAU Service Layer
PL : Presentation Layer - PS : Presentation Services

2 : supposed to provide something similar, but no relevent information is available.

7.3. Advantages and disadvantages of those Architectures

7.3.1. SNA

7.3.1.1. Remark :

To some people, it seems that increasing distribution of the control function within communication networks and distributed networks will lead to 'conversion' of SNA into a range of other products which may or may not carry the same trade name (something like 'DNA' aligned with OSI Reference Model) [DC IBM].

7.3.1.2. Advantages :

1. High degree of communication/application independence

   IBM's SNA separates application programs from communication system programs and provides the much needed independence from each other. This separation is useful when communication common carriers come up and develope. It should be born in mind, however, that this advantage is not unique to SNA.

2. Transparent network

   Any bit stream is allowed and the end user is not concerned with network topology, route selection, or media used.

3. SNA, being centralised, facilitates the provision of important services such as safeguard for data integrity, end-to-end data security, confidentiality, user authentication, encryption,...

4. Terminal access to all application and several computers. But network functions requested are complex.

7.3.1.3. Disadvantages

1. One level network : each computer hosts application programs and network control modules which involve an extra load in the computer. In counterpart, this implementation is cheaper for those who cannot afford a separated private self-constituting network.

2. Locks user into IBM

   Some people have accused SNA as being an IBM strategy in the marketing approach to teleprocessing which is designed to sell IBM equipment.

   Present day users have become increasingly sophisticated and they demand their independence and flexibility from their computer networks, including choice of equipment. This make SNA viewed with suspicion (IBM trying to control an emerging market; SNA becoming a standard de facto for IBM has a powerful market position) because locking the user into IBM.

3. SNA too expensive, too complex, and too dependent on main frame computers

   - a) to some people the fact that SNA is a one level network architecture : network control programs and application programs reside into the same computers. Network and network control is not a whole independent from host systems (mainframe computers); this involves a certain overhead of the mainframe computers.

   - b) enormous complexity of SNA, of computers supporting SNA (this make not easy to understand price performance trade-offs).

   - c) SNA documentation is considered extremly complex.

4. Above Path Control, SNA defined Protocols may handle and work on other headers than their owns. This opposes the layering principles defined in OSI Reference Model, and makes Protocol functions hard to understand.

5. Network control and management are highly centralized. This hierachical topology makes complex, for instance, the session binding process between two L.Us. located in different domains (fourteen control messages are required; see [Tanenbaum 81] page 377).

6. Reconfiguration process, in case of addition of a

new node, is not easy and non automatized (like in DNA for instance).

7.3.2. DSA

7.3.2.1. Advantages :

1. Two level network architecture

   The host nodes are unloaded of communication tasks which are handled by the network processors. The communication network constitute an independent whole in itself and does not rely on hosts for its correct operation.

7.3.2.2. Disadvantages :

1. This two level approach is meanwhile more expensive than a one-level one due to the extra cost of network processors not needed in the preceedings architectures.

7.3.3. DNA

7.3.3.1. Advantages :

1. High degree of communication/application independence.

2. Distributed network architecture

   No one computer does network control for the others (like SSCP within SNA), there is no master/slave distinction and no identifiable central control point.
   Network control steps are managed in each node more or less symetrically. This allows, in principle, a wide range of topologies to be implemented(point-to-point, star, meshed, hierarchical, non hierachical topologies).

   This method sets bounds to the size reachable by the network. For large network of hundred of computers, network management functions should be more centralised (routing, addressing). An upper hierachical network layer could be set up.

3. DNA independence of the internal characteristics of Digital computers and their Operating Systems. Specific modules interface between O.S. and DNA environments.

4. Access for every application to every application/resource wherever in the network.

5. Automatic reconfiguration in case of node introductions or failures.

6. High availability :

   Routing, based on Datagram service, allowing automatic rerouting in case of route parameter modifications or route failure. DECnet networks can be configured to maintain operation even if a subnet of lines or nodes fail. Because maintenance functions are highly ditributed, DECnet network can recover from operator error unless the error is extreme [DNA GD].

7.3.3.2. Disadvantages :

1. One level network

   Each computer hosts application programs and network control or/and communication modules. These latters involve a certain overhead of the computer. This is not always suitable, but in counterpart, it is cheaper than to have to pay for extra processors dedicated to networking. Meanwhile it is to note that a self-constituting newtork could be set up, that serve data processing host computers, relieving them from networking concern.

2. No features like encryption or data integrity/confidentiality throughout the network seems to be provided. Only password mechanism exists to secure data and network accesses.

3. Does not provide so much features than SNA or DSA at Session level.

4. In spite of the DNA independence from various Operating Systems, the Session Control is the point where DNA relates to the node resident O.S. Some modules of this layer depend then on the computer they are running into.

8. CONCLUSIONS :

The overall comparisons end here. They are not complete and we don't intend to disguise that fact. A longer delay and much information, discusses, meeting should have been necessary to realize such an approach. It is even doubtful that this goal is reachable, because of the continual enhancements in networking area.

We recall that our aim was to contribute to an approach of the moving Network Architectures field, making some concepts clearer, giving a synthetical overview of what is done in a part of the Computer Communications world.

More time and much information would have allowed us

- To terminate the overall descriptions of architectures.
- To go further in the comparisons between the various architectures :
  - By forwarding in what concerns their advantages and disadvantages,
  - Throught two-by-two comparisons aiming to underline major and minor differences between those architectures.
- To pursue the Protocol Cuts analysis, going deeper in cross comparisons.
- To meet people (contructors, users, member of ECMA, ...) for advice and information.
- To use some of these communication software in order to acquire a better understanding of their operation and function.

S E C O N D E P A R T I E

IMPLEMENTATION d'UNE MAQUETTE DES 3 COUCHES SUPERIEURES DU MODELE DE REFERENCE OSI DE L'ISO.

-=-=-=-=-

- - TABLE DES MATIERES - -

	Pages
1. OBJECTIF DU TRAVAIL	1
2. STRUCTURE GENERALE DU MODELE	2
2.1. Décomposition en couches	
2.2. Relations inter-couches et notions de point d'accès	
2.2.1. Définitions	
2.2.2. Mapping des différents niveaux d'adresse	3
2.2.3. Identifieur et adressage dans les 4 couches supérieures	4
2.3. Choix d'architecture standard de couche	6
2.3.1. Description	
2.3.2. Justification du choix d'interfaces entre couches	8
3. HYPOTHESES DE TRAVAIL	9
4. NIVEAU PRESENTATION	12
4.1. Description générale	
4.2. Choix liés à la maquette et services sélectionnés	
4.3. Spécifications fonctionnelles des primitives	13
5. NIVEAU SESSION	15
5.1. Description générale	
5.2. Concepts utilisés	
5.2.1. Terminaison de connexion-session	
5.2.2. Concepts de " jetons "	16
5.3. Services choisis	17

5.4. Notions liées aux spécifications des primitives 18

5.4.1. Mapping transport-address
5.4.2. Contrôle de flux 19

5.5. Description formelle du protocole de session 20

5.5.1. Etats
5.5.2. Conditions 21
5.5.3. Evénements 22
5.5.4. Diagramme d'états & transitions 23
5.5.5. Tables de transitions 24

5.6. Spécifications fonctionnelles des primitives du niveau 32

5.7. Spécifications fonctionnelles des primitives de l'interface Session/Transport 45

5.8. Spécifications fonctionnelles du moniteur d'interruption Message-Arrival 47

6. NIVEAU TRANSPORT 48

6.1. Description générale

6.2. Choix liés aux hypothèses de travail 49

6.3. Services choisis

6.4. Description générale des primitives de service choisies

6.5. Notions liées aux spécifications des primitives 50

6.5.1. Contrôle de flux

6.6. Description formelle du protocole de transport

6.6.1. Etats
6.6.2. Evénements 51
6.6.3. Diagramme d'états et transitions 53

6.7. Spécifications fonctionnelles des primitives du niveau 54

6.8. Spécifications fonctionnelles des primitives de l'interface Transport/Réseau 59

6.9. Spécifications fonctionnelles du moniteur d'interruption Packet-Arrival 61

6.IO. Hiérarchie des primitives du niveau transport de la maquette

7. PROBLEMES RELATIFS A LA MISE EN OEUVRE 62

7.1. Simulation des interruptions

7.2. Flux de messages et commandes au niveau Session 63

7.2.1. Problèmes liés à un seul flux pour le niveau Session

8. DISCUSSIONS DE MODIFICATIONS DE LA MAQUETTE

1. OBJECTIF DU TRAVAIL.

L'objectif, fixé au départ de ce travail pratique, est de réaliser une maquette, une implémentation simplifiée, des 3 couches supérieures du Modèle de Référence OSI. C'est-à-dire les couches Présentation, Session et Transport.

Le langage choisi est le Pascal Unix VU (Vrije Universitat, Amsterdam). Ce choix ainsi que la version disponible sur le PDP, imposeront des limitations quant à l'implémentation (inexistences de mécanismes de synchronisation entre processus parallèles et/ou concurrents.

De même un certain nombre d'hypothèses de travail seront posées et discutées, quant à leurs limitations et leurs implications.

Dans une première partie nous aborderons la structure choisie pour le modèle et les hypothèses y attenant. Ensuite nous décrirons les différentes couches par le biais de leurs primitives et procédures de contrôle. Enfin une discussion des améliorations, modifications d'hypothèses et implications termineront ce travail.

Remarque : Dans la suite les références bibliographiques seront notées (° référence).

2. STRUCTURE GENERALE DU MODELE.

2.1. Décomposition en couches.

La structure de communication de la fig. 1 fournit aux entités de la couche Application le support nécessaire à leur dialogue.

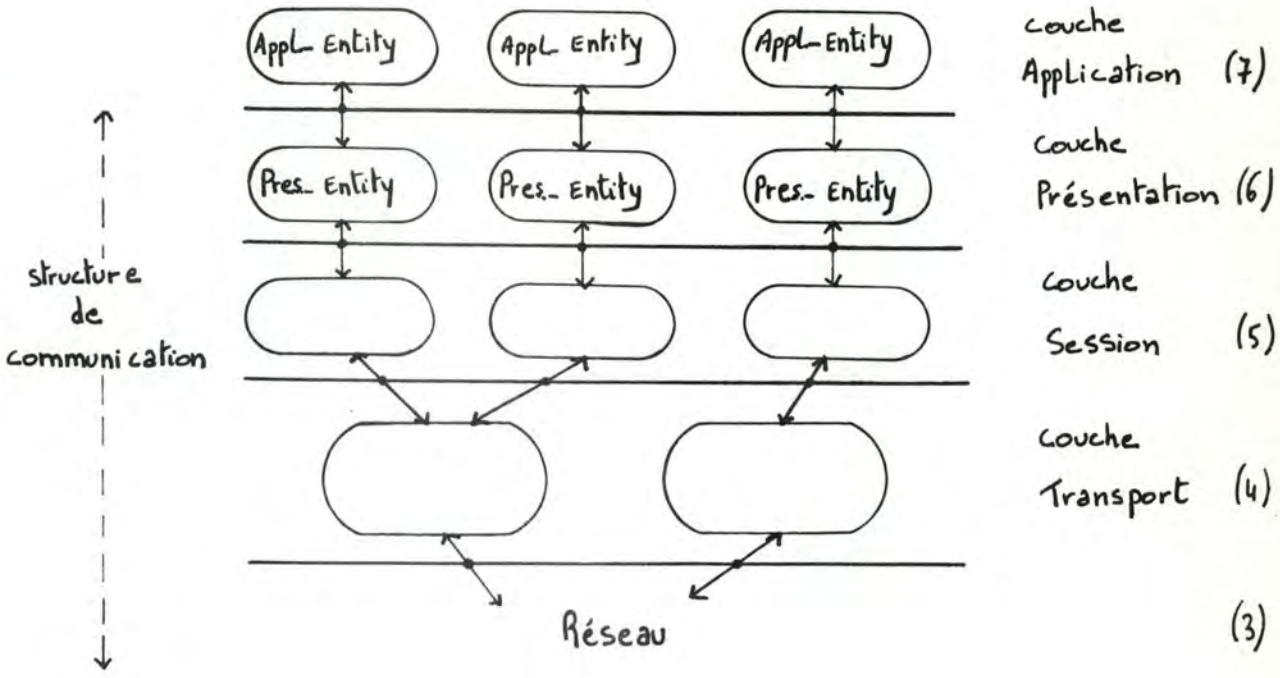


Figure 1

2.2. Relations inter-couches et notions de point d'accès.

2.2.1. Définitions (° OSI RM).

- (n)adresse : abréviations de " adresse du point d'accès au (n)service ".

- <u>adresse du point d'accès au (n)service</u> : identifiant d'un point d'accès particulier. Cette adresse est utilisée par une entité de niveau (n + 1) pour désigner le point d'accès lorsqu'elle requiert une (n)connexion à la couche (n). Une (n)entité est associée à ce point d'accès et preste les services requis.

- <u>point d'accès au (n)service</u> : association entre une (n)entité et une (n + 1) entité lorsqu'un (n)service est requis par cette (n + 1) entité.

2.2.2. <u>Mapping des différents niveaux d'adresse</u>.

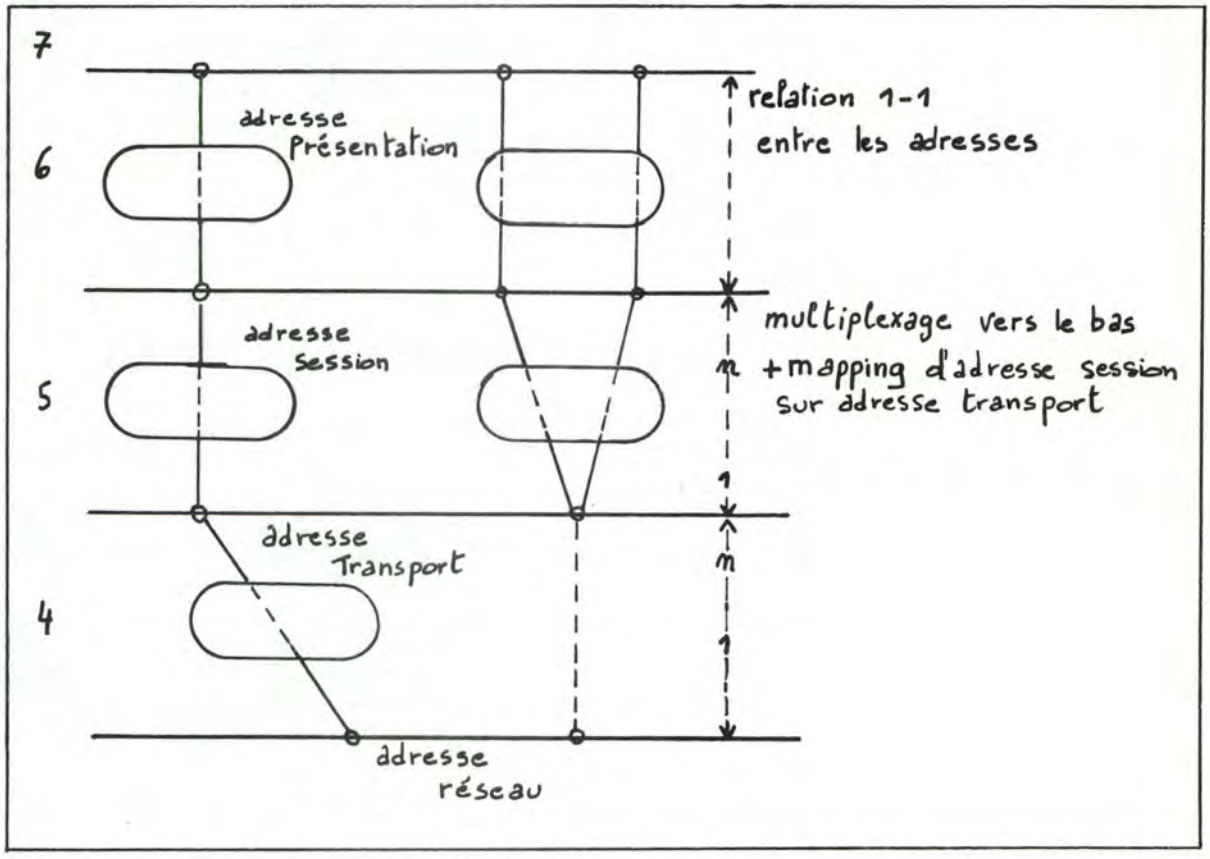


Figure 2 : mapping des différents niveaux d'adresse

2.2.3. Identifieur et adressage dans les 4 couches supérieures (fig. 3):

1. Une entité-application de nom " Alpha A " (connue de son niveau Présentation par son adresse-présentation L 1) demande à l'entité-présentation la servant d'établir une connexion-présentation avec une entité-application de nom " Beta B " connue par son adresse-présentation B 2 .

2. Le niveau Présentation ne supportant pas de fonction d'adressage, la demande est passée à l'entité-session " Alpha " servant l'entité-présentation.

3. L'entité-session " Alpha " identifie l'adresse-transport B (nous supposons ici un adressage hiérarchique) à utiliser pour atteindre l'entité-session, distante, qui sert l'adresse-session B 2 (et par cette adresse, ou porte, un niveau Présentation).

4. L'entité-session " Alpha " demande alors, à son entité-transport " a 7 " , via une adresse-transport " L ", d'établir une connexion-transport avec l'adresse-transport distante " B ".

5. L'entité-transport " a 7 " détermine l'adresse-réseau à utiliser pour atteindre l'entité-transport " b 7 " servant l'entité-transport " B ". Supposons que cela se fasse par mapping, et que l'adresse trouvée soit " 40 ".

6. En supposant que la connexion Réseau soit établie, la demande d'établissement de connexion-transport entre L et B est envoyée à " b 7 ".

7. Lorsque la connexion-transport est établie, l'entité-session " Alpha " envoie à l'entité-session " Beta " une demande d'établissement de connexion-session entre les adresses-session " L 1 " et " L 2 ".

8. Lorsqu'elle est établie, le niveau Présentation la répercute au niveau Application. Le dialogue peut alors se dérouler.

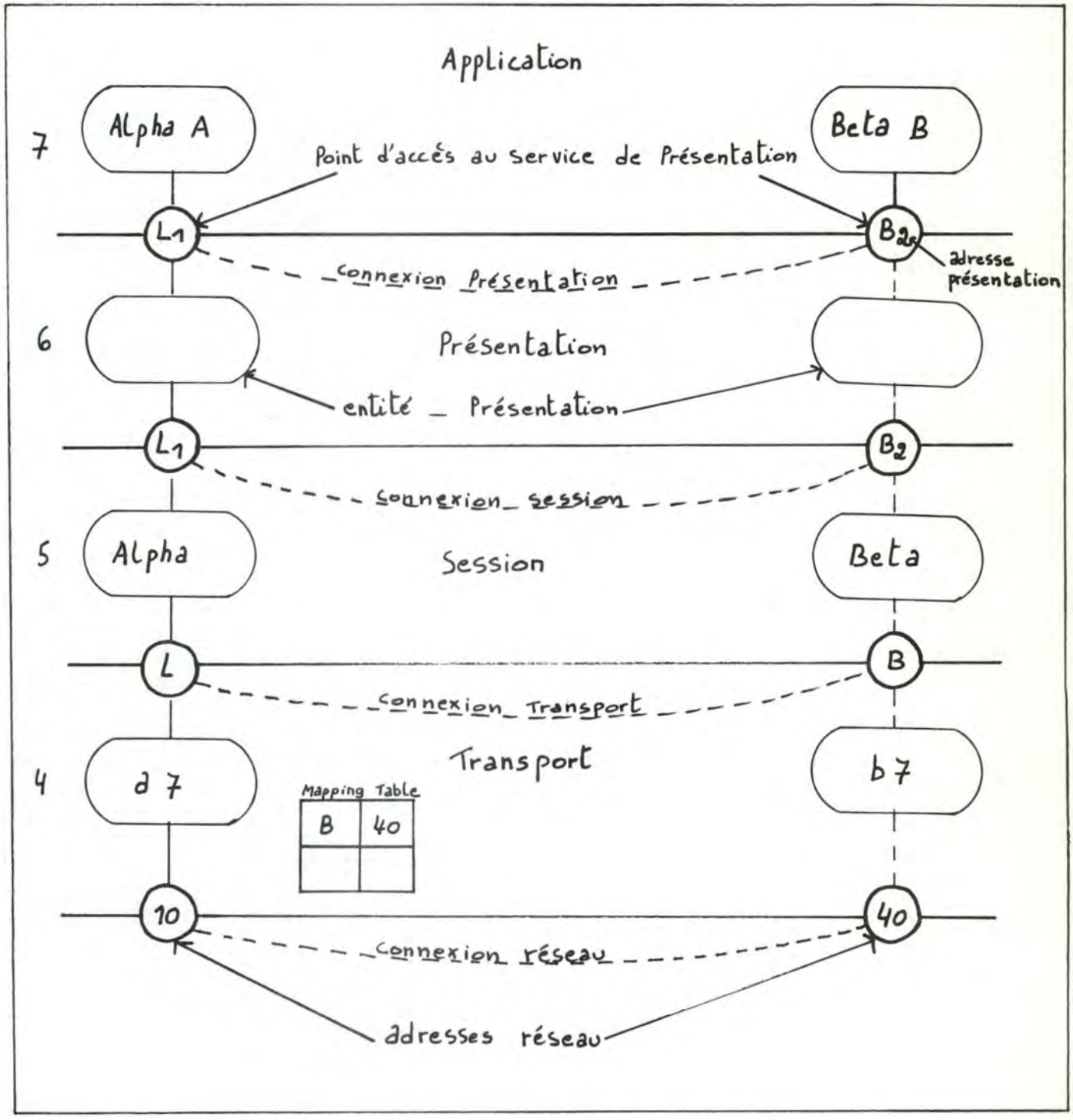


Figure 3 : identifiants et adressage

2.3. Choix d'architecture standard de couche.

2.3.1. Description.

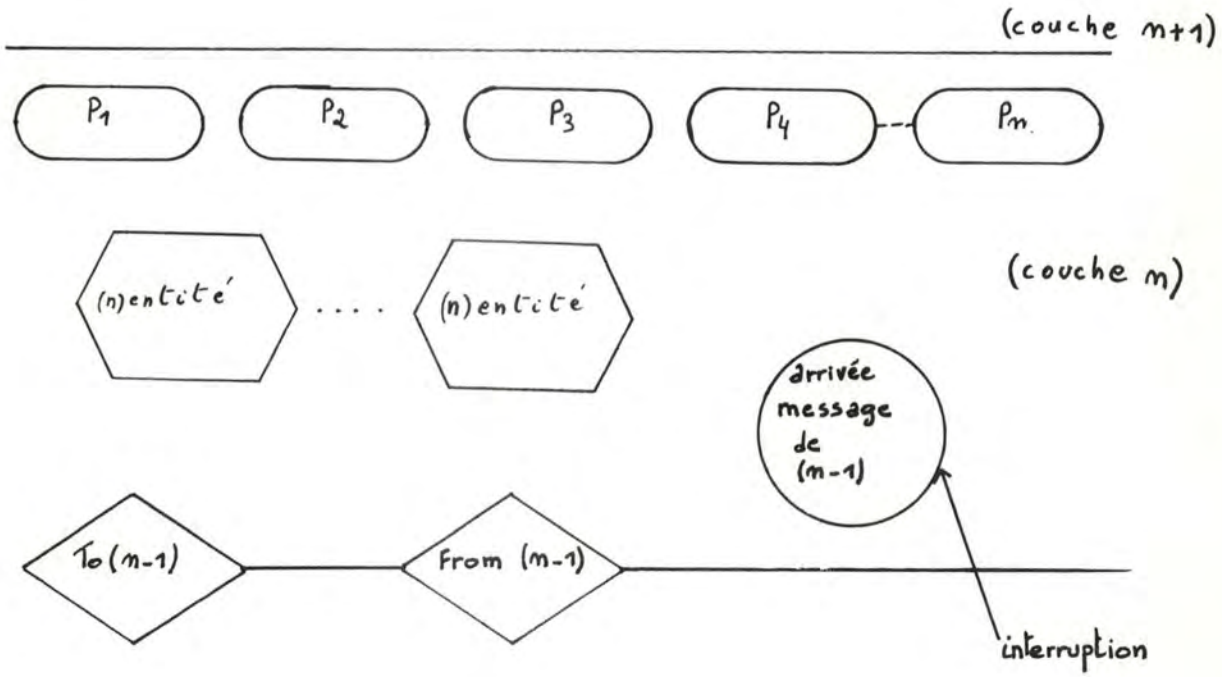


Figure 4 : Architecture de couche

Une couche (n) est composée d'(n)entités fournissant un (n)service à la couche (n + 1).

Pour fournir ce (n)service, les (n)entités font appel à des Primitives (P1 à Pm) réalisant une fonction, un service déterminé (transmission/réception de données, ouverture/fermeture de (n)connexion).

Ces Primitives, afin de réaliser leur fonction, utilisent les services de la couche (n - 1). L'interfaçage entre les entités des deux couches adjacentes, se fait via les procédures d'interface To (n - 1) et From (n - 1). Ces interfaces règlent les problèmes de communication pouvant se présenter (synchronisation, dialogue entre processus ...).

Dans l'exécution de leur fonction, les (n)Primitives peuvent être amenées à attendre la réalisation d'événements. L'entité impliquée sera alors mise au repos. Un moniteur d'interruption Arrivée-Mess (n - 1) prend en charge la surveillance d'occurrence d'événements et le réveil des entités intéressées (ainsi que la reprise du programme de la Primitive) fig. 5 .

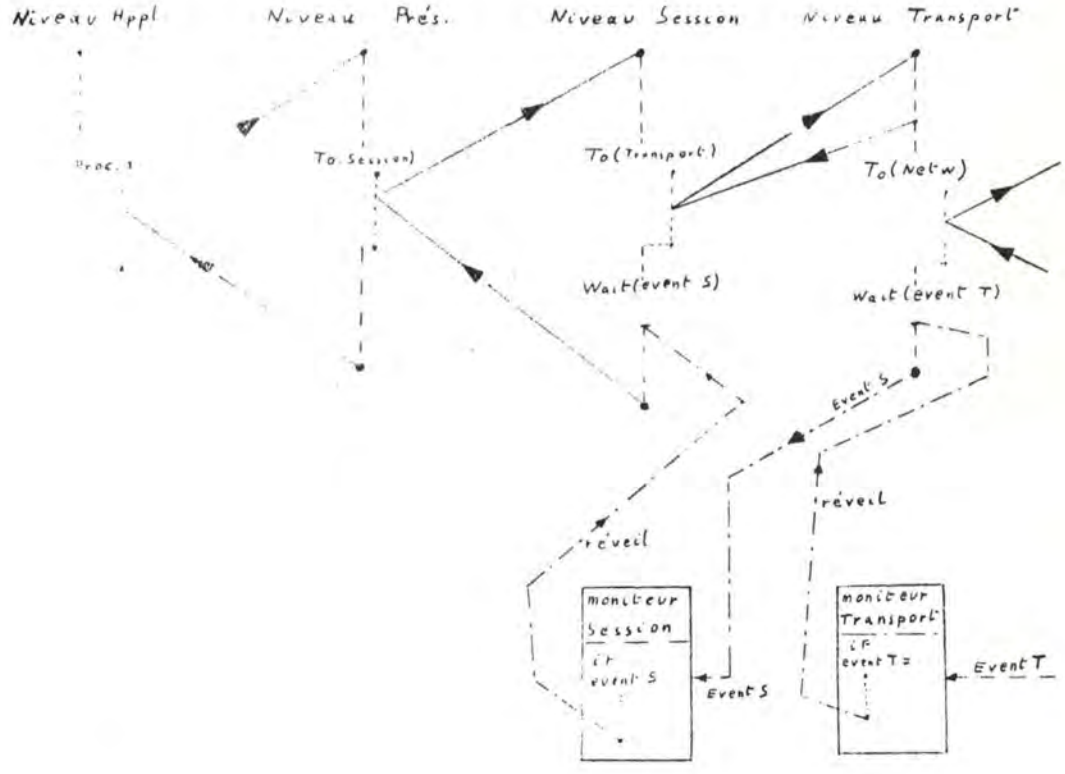


Figure 5 : enchaînement des appels.

Remarque :

Une (n - 1)entité peut servir plusieurs (n)entités. La relation n'est pas obligatoirement 1 - 1 (voir fig. 2).

2.3.2. Justification du choix d'interface de communication entre couches adjacentes:

1. Cacher l'information, c'est-à-dire, la façon dont est implémenté le niveau inférieur. Et ce, afin de faciliter les modifications de ce niveau et limiter les répercussions au niveau supérieur au seul module d'interface.

2. Développement séparé des couches après spécification des services fournis par la couche inférieure et des services à fournir à la couche supérieure.

Remarque :

Ces modules peuvent être des goulots d'étranglement pour le dialogue entre couches adjacentes. Ils ne doivent donc faire que le strict minimum : passer l'information.

Pour éviter ce goulot, plusieurs modules pourraient être actifs à un niveau.

3. HYPOTHESES DE TRAVAIL.

Dans le but de faciliter le travail de réalisation, la présentation et la compréhension de cette maquette, nous sommes amenés à poser les hypothèses suivantes :

1. Des hypothèses seront faites en raison du langage choisi (Pascal Unix V U - Vrije Universitat - Amsterdam -), et afin de ne pas déplacer un problème de modélisation d'architecture vers des problèmes relatifs à un Système d'Exploitation :

   - gestion de processus concurrents et/ou parallèles
   - gestion de mécanismes d'interruptions multiples
   - gestion de mécanismes de synchronisation de processus
   - gestion de ressources
   - multi-tasking

   Un modèle englobant ces différents problèmes serait, bien entendu, souhaitable. Mais, étant donné le temps imparti et les difficultés que cela poserait, nous choisissons des bases de travail plus restrictives et non nécessairement réalistes.

2. Simplicité de réalisation.

   Un modèle plus réaliste devrait tester la validité de tous les paramètres d'appel des primitives (droit d'appel,...), gérer des collisions possibles de messages, gérer les interruptions, des attentes de conditions. Nous ne le faisons pas.

3. Nous supposons l'existence de 2 primitives-système :

   - SLEEP : endort le processus appelant.
   - WAKEUP : réveille le processus impliqué.

4. Nous nous basons sur un service Transport dérivé de celui décrit dans (° Tanenbaum 81) p. 328 - 335.

5. Unicité de l'utilisateur du niveau Application dans un Système. Celui-ci est sensé simuler les processus de niveau Application. Il dispose de l'enveloppe de communication (structure non réentrante) considérée comme étant une librairie de modules de communication.

   Deux utilisateurs dans deux systèmes auront donc, chacun à leur disposition, une enveloppe de communication particulière.

6. Les moniteurs d'interruption sont ininterruptibles une fois activés. Ils déroulent leurs procédures jusqu'à leur fin logique.

   Seul un moniteur d'interruption peut appeler la primitive WAKEUP.

7. Séquentialité des enchaînements de procédures au travers des différentes couches : deux actions ne peuvent être réalisées en parallèle.

   Exemple :

   A - demande d'une connexion-session X
   B - demande d'une connexion-session Y

   B ne pourra commencer à s'exécuter qu'après la fin de A.

8. Mécanisme de contrôle de flux basé sur le crédit (autorisation à émettre) plutôt que sur le mécanisme de " Sliding-Window " (fig. 6). Et ce, afin de ne pas devoir gérer des buffers de réception multiples au sein des différentes couches. Nous aborderons le problème de buffers multiples en section 8.

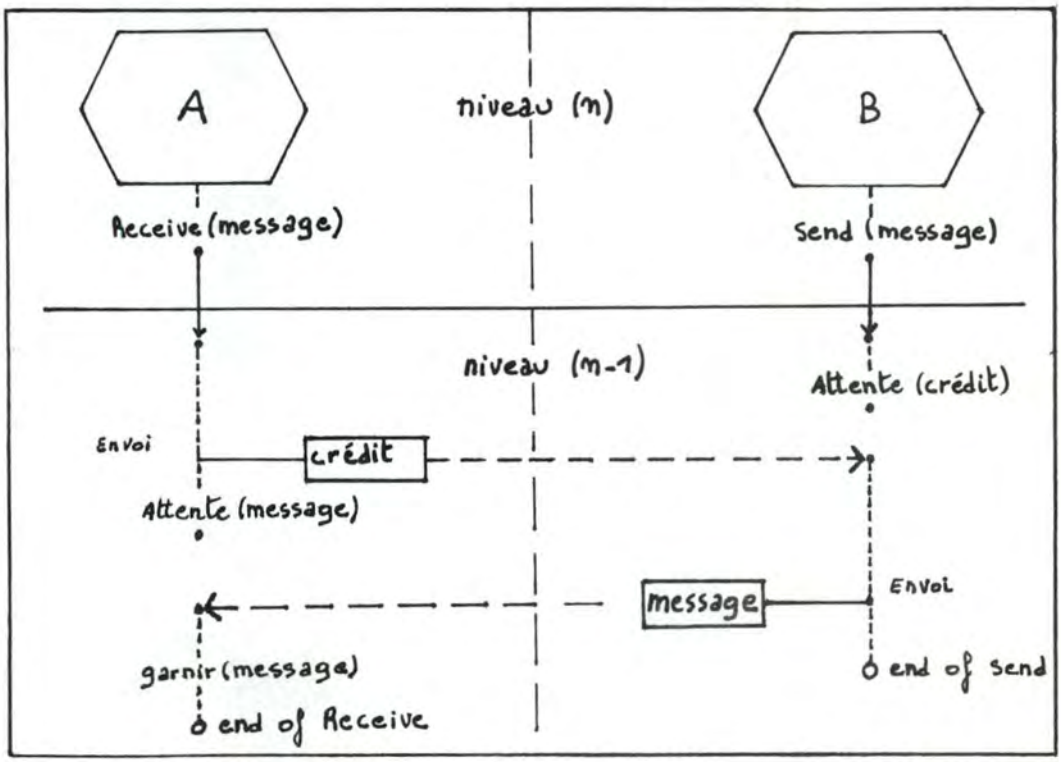


Figure 6 : mécanisme de crédit

9. Les couches ne sont pas implémentées par des processus concurrents mais constituent un seul processus.

10. Les descriptions formalisées se feront sous forme de diagramme d'états et transitions et/ou de tables de transitions.

4. NIVEAU PRESENTATION.

4.1. Description générale (° ECMA DPP)

Le rôle du protocole de présentation est de permettre un échange standardisé et significatif d'informations entre deux entités application, dans le contexte de l'Interconnexion de Systèmes Ouverts (OSI), et ce, indépendamment de chaque système spécifique.

Les services fournis par ce niveau peuvent être divisés en deux catégories :

- services relatifs à la présentation des données.
- services de contrôle du dialogue qui sont le reflet de ceux du niveau Session.

4.2 Choix liés à la maquette et services sélectionnés.

En vertu des hypothèses de simplicité et compréhension, et du manque de temps nécessité par une étude plus approfondie,

- ce niveau assure la seule conversion caractères → entiers / entiers → caractères.
- ce niveau accepte des chaînes de caractères de longueur maximum " maxsdu " (évite de reproduire un mécanisme de segmentation de message déjà existant au niveau Session).
- Il passe, de façon transparente, les commandes au niveau Session.

Les interfaces Application/Présentation et Présentation/Session ne sont pas réalisées, par manque de temps. Le niveau Application fait donc directement appel aux primitives du niveau Présentation.

Les services fournis par ce niveau sont donc nuls, mis à part les conversions de code.

4.3. Spécifications fonctionnelles des primitives.

```
1.      convi ( charm      :  sduchar ;
                lcharm     :  integer ;
                var intm   :  sdutype ) ;
```

a) spécification

Réalise la conversion de la chaîne de caractères contenue dans le buffer référencé par ' charm ' en une chaîne de valeurs ASCII contenue dans ' intm '.
La longueur de ' charm ' est contenue dans ' lcharm '.

```
2.      convc ( intm       :  sdutype ;
                lintm      :  integer ;
                var charm  :  sduchar ) ;
```

a) spécification

Réalise la conversion de la chaîne de valeurs ASCII contenue dans le buffer référencé par ' intm ' en la chaîne de caractères équivalente qui sera placée dans ' charm '.
La longueur de la chaîne ASCII est ' lintm '.

```
3.      P-connect ( init, accp        : sessaddress ;
                    tokenset          : tokenrec ;
                    var res, status   : integer ) ;
```

Même spécification que pour S-connect (voir section 5.6).

```
4.      P-activate ( init              : sessaddress ;
                     var acp           : sessaddress ;
                     tokenset          : tokenrec ;
                     var res, status   : integer ) ;
```

Même spécification que pour S-activate.

```
5.      P-data     ( sessid            : sessidtype ;
                     userdata          : sduchar ;
                     datalg            : integer ;
                     var res, status   : integer ) ;
```

Même spécification que pour S-data excepté le type du message ' userdata ' qui est, ici, une chaîne de caractères.

6. Les autres primitives ont toutes les mêmes spécifications que celles du niveau Session mis à part le type d'éventuels messages (voir 5 ci-dessus) et le préfixe P qui remplace S.

5. NIVEAU SESSION.

Les services assurés par le niveau Session de la maquette constituent une version simplifiée de ceux décrits dans (° ECMA SP).

5.1. Description générale.

Le rôle du protocole de Session et des services de ce niveau, est d'améliorer les services fournis par le niveau Transport.

Les principales caractéristiques des services du niveau Session sont :

- transparence et fiabilité du transfert de données.
- gestion et organisation du transfert de données.
- synchronisation du transfert de données.

La transparence et la fiabilité sont directement dérivées des services du niveau Transport. Les deux autres caractéristiques sont des " valeurs ajoutées " par le protocole de Session.

L'organisation du transfert est plus particulièrement concernée par l'établissement et la terminaison ordonnée de connexions-session, et, par la gestion et la structuration des flux de données échangées dans une connexion-session.

La synchronisation est, quant à elle, concernée par les problèmes relatifs aux déficiences et incertitudes dues aux délais de transmission (° ECMA SP).

5.2. Concepts utilisés.

5.2.1. Terminaison de connexion-session.

Trois types sont possibles :

- terminaison anormale (S-abort).
- terminaison non disruptive (S-release).
- terminaison négociée (S-release).

La première peut engendrer des pertes de données. Les deux autres ne causent aucune perte de données. La différence entre la seconde et la troisième est que dans la dernière forme, la session acceptante (recevant le message) peut refuser de terminer la connexion et stipuler sa volonté de poursuivre l'échange.

5.2.2. Concepts de " jetons " (Tokens).

Ces " jetons " fournissent, aux utilisateurs des services du niveau Session, un mécanisme de contrôle dynamique et non ambigu, ainsi qu'un droit exclusif à initialiser certaines fonctions.

Chaque " jeton " a une durée de vie égale à celle de la connexion-session. Il est déterminé et assigné lors de l'initialisation de cette dernière.

Cette assignation est faite à l'un ou l'autre des utilisateurs des services session. La propriété du " jeton " peut cependant changer durant le cours de la connexion-session selon les modalités de l'échange de données.

Seuls les " jetons " déterminés lors de l'initialisation peuvent intervenir durant la connexion.

Dans notre maquette nous utiliserons deux types de " jetons ":

1. jeton de droit d'émission de données (data token).
2. jeton de terminaison négociée (terminate token).

Le premier est défini pour les liaisons logiques ' une voie bidirectionnelle à l'alternat ' (TWA) et ' une voie monodirectionnelle ' (OW). Il détermine le " tour de parole " entre les deux utilisateurs des services session. Il n'a pas de sens pour les liaisons " full-duplex ".

Le deuxième détermine la possibilité de terminer une connexion-session de façon négociée ou non.

Par exemple, si seul le terminate token est défini, la communication sera " full-duplex " (TWS), et seul l'utilisateur auquel il est assigné, pourra initialiser une demande de terminaison négociée de la connexion-session.

5.3. Services choisis (1).

1. Etablissement et terminaison de connexion-session : S-connect, S-release, S-disconnect, S-not-Finished, S-abort.

2. Transfert de données (réception/émission) : S-data, S-expedited, S-receive.

3. Synchronisation (échange de " jetons ") : S-token-give, S-please.

Ces services seront décrits par les " primitives " qui leur sont associées.

Nous nous sommes volontairement restreints à ce sous-ensemble de services afin de simplifier la réalisation et la compréhensibilité de la maquette. Il correspond au Subset A, défini dans (° ECMA SP) p. 73, enrichi des primitives d'échange de " jetons ".

---

(1) : pour une description détaillée de ces services, reportez-vous au document (° ECMA SP).

5.4. Notions liées aux spécifications des primitives.

5.4.1. Mapping, " initiator " et " acceptor address ", transport address.

La couche session définit un certain nombre d'adresses-session (fig. 7) via lesquelles s'établit une connexion-session. Les adresses-session fournissent le moyen, pour les entités (utilisateurs) du niveau supérieur, de s'appeler, de se référencer entre elles. Une connexion-session est constituée par 2 adresses-session et la liaison logique qui les relie. Le tout est identifié par un numéro logique de connexion.

La liaison logique est réalisée par l'intermédiaire d'une connexion-transport fournie par la couche Transport.

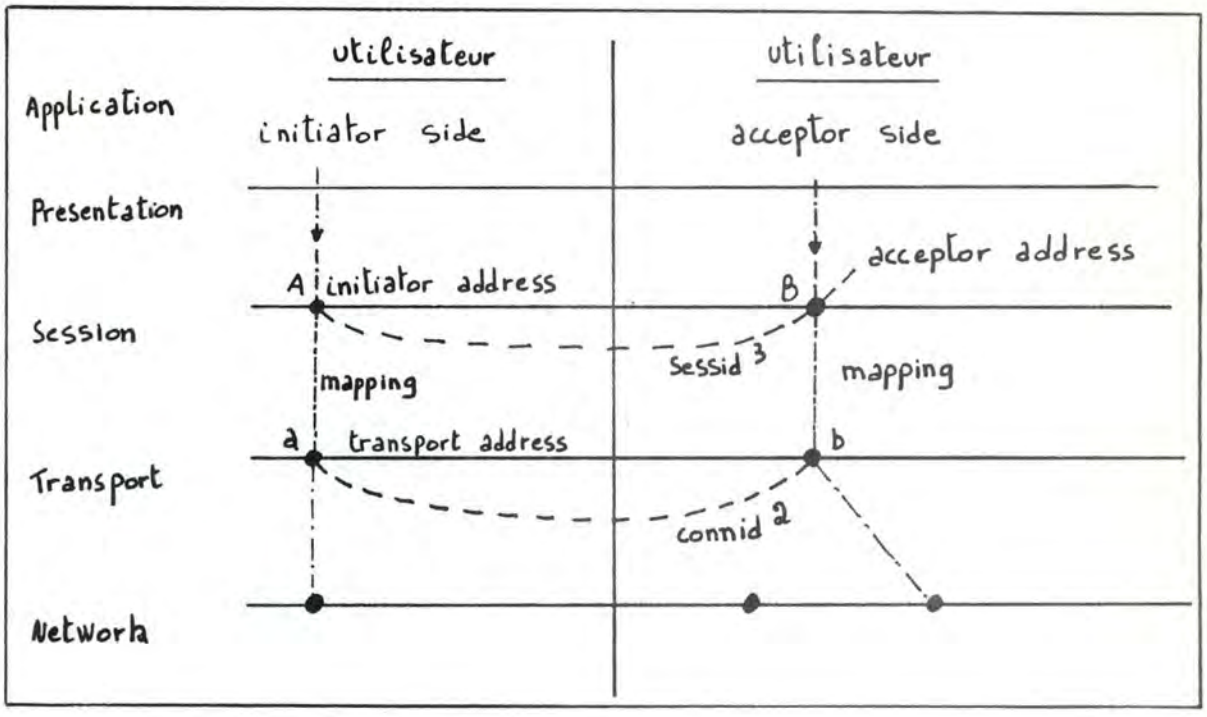


Figure 7

Le Mapping d'une adresse-session sur une adresse-transport se fait par calcul et à l'aide d'une table de mapping.

5.4.2. Contrôle de flux.

Deux flux sont distingués sur une connexion-transport (fig. 8).

- le flux des données.
- le flux des commandes.

Le flux des données est régulé par un mécanisme de crédit d'émission (voir hypothèses).

Le flux des commandes est indépendant du premier et n'est pas soumis à un mécanisme de régulation.

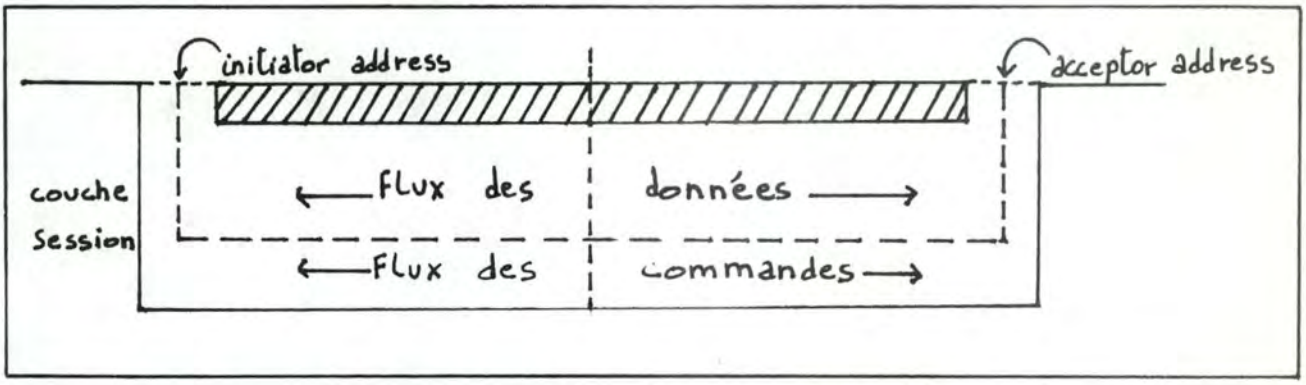


Figure 8 : Flux d'une connexion-Session

5.5. Description formelle du protocole de session.

5.5.1. Etats.

- Unactivate : état neutre correspondant à " aucune session en cours ".

- Unconnected : état lié à la réception d'un message " Abort ". Il précède le retour à l'état neutre " Unactivate ".

- Wait-cn-msg : état lié à l'attente d'un message de demande de connexion-session (CN) suite à un S-activate.

- Wait-estab-rsp : état lié à l'attente du résultat de la négociation d'une demande de connexion reçue (REFUSE ou ACCEPT).

- Wait-acc-msg : état lié à l'attente du message de refus ou d'acceptation de connexion-session suite à l'envoi d'une demande de connexion (CN).

- Opendt : état lié à une connexion-session ouverte (en cours) avec ' data token ' défini et assigné à l'utilisateur local.

- Openndt : état lié à une connexion-session ouverte avec ' data token ' défini et assigné à l'utilisateur distant.

- Open : état lié à une connexion-session ouverte de type ' Full-duplex ' (data token non défini).

- Wait-disc-msg : état lié à l'attente d'un message d'acceptation de terminaison de connexion, ou, de refus de terminaison de connexion (dépendant de la définition du ' terminate token ') suite à l'envoi d'une demande de fin de connexion (FN).

- Wait-rlse-rsp : état lié à l'attente de réponse (acceptation de terminaison ou refus) suite à la réception d'un message de demande de fin de connexion.

5.5.2. Conditions.

Table 1 : Conditions

Conditions	Meaning
DT	DATA TOKEN my side
^DT	DATA TOKEN not my side
	END-DU-TOKEN not my side
TR	TERMINATE-TOKEN my side
^TR	TERMINATE-TOKEN not my side
DEF (xx)	TOKEN XX is defined
^DEF (xx)	TOKEN XX is not defined

5.5.3. Evénements.

1. Messages.

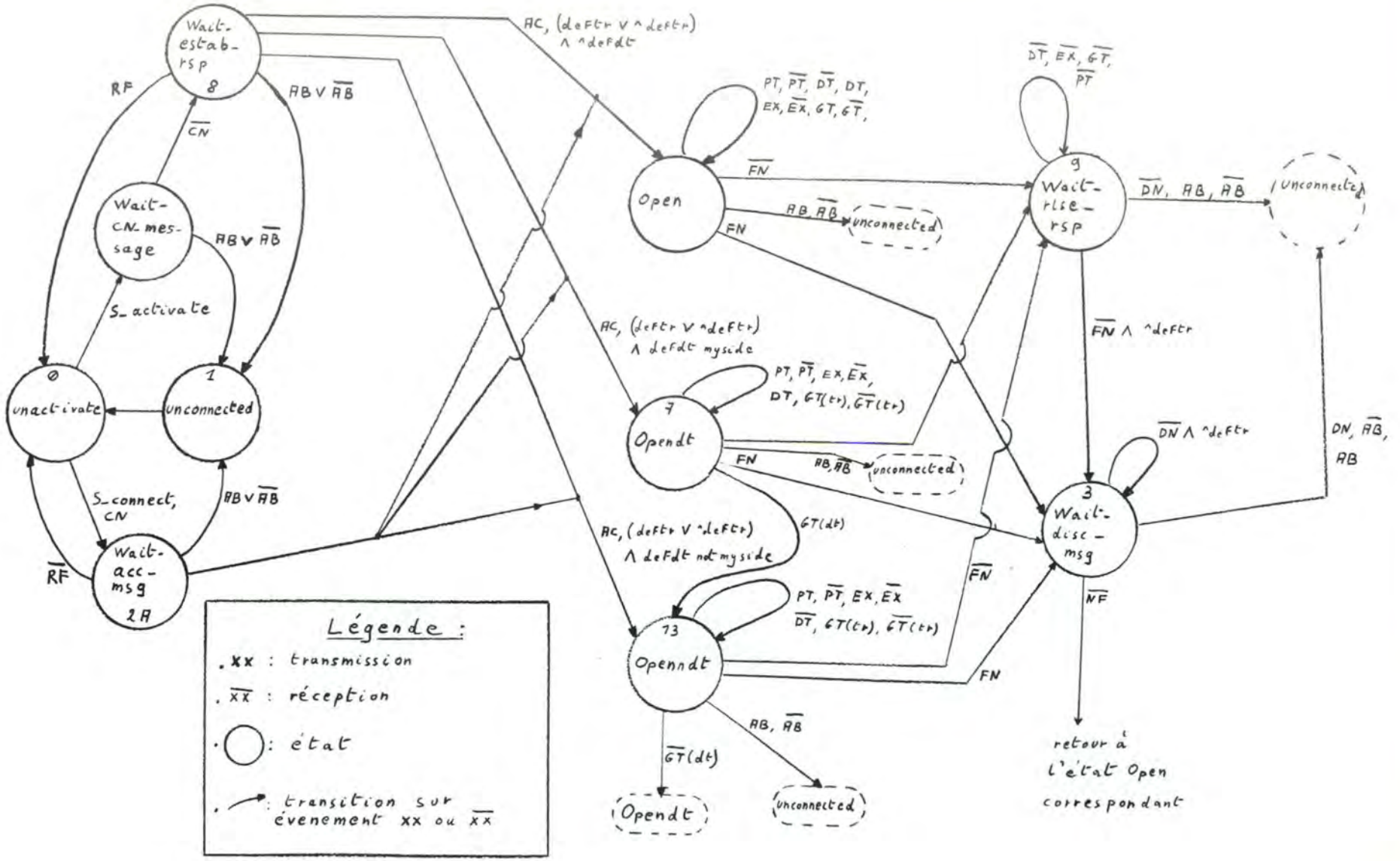
Table 2a : messages

CONNECT	CN
ACCEPT	AC
REFUSE	RF
FINISH	FN
NOT FINISHED	NF
DISCONNECT	DN
ABORT	AB
DATA TRANSFER	DT
EXPEDITED	EX
CREDIT	CDT
PLEASE TOKENS	PT
GIVE TOKENS	GT

2. Primitives.

S-activate.

5.5.4. Diagramme d'états & transitions (fig. 9).



5.5.5. Tables de transitions.

1. Codage des états.

State Code	State Description
STA 1 (unconnected)	Unconnected
STA 2A (wait.acc.msg)	waiting for ACCEPT message
STA 3 (wait_disc_msg)	waiting for DISCONNECT message
STA 7 (Opendt)	Idle DATA TOKEN my side
STA 8 (wait_estab_rsp)	waiting for ESTABLISH response event
STA 9 (wait_rlse_rsp)	waiting for RELEASE response event
STA 13 (Opennd t)	Idle DATA TOKEN not my side

Table 2b. Event code

Event Code		Event Description
EVE 1	(CN)	ESTABLISH request event
EVE 2	(AB)	ABORT request event
EVE 3	(FN)	RELEASE request event
EVE 4	(DT)	TRANSFER request event
EVE 6	(EX)	EXPEDITED request event
EVE 8	(GT)	GIVE TOKENS request event
EVE 9	(PT)	PLEASE TOKENS request event
EVE 11	($\overline{CN}$)	CONNECT incoming message event
EVE 12	($\overline{AC}$)	ACCEPT incoming message event
EVE 13	($\overline{FN}$)	FINISH incoming message event
EVE 14	($\overline{DN}$)	DISCONNECT incoming message event
EVE 15	($\overline{DT}$)	DATA TRANSFER incoming message event
EVE 18	($\overline{EX}$)	EXPEDITED incoming message event
EVE 21	($\overline{GT}$)	GIVE TOKENS incoming message event

Table 2b. continued

EVE 23	$(\overline{PT})$	PLEASE TOKENS incoming message event
EVE 31	$(\overline{AB})$	ABORT incoming message event
EVE 33	$(\overline{NF})$	NOT FINISHED incoming message event
EVE 25	(AC)	ESTABLISH response event
EVE 26A	(FN)	RELEASE AFFIRMATIVE response event
EVE 26B	(NF)	RELEASE NEGATIVE response event
EVE 101		ESTABLISH indication event
EVE 102		ABORT indication event
EVE 103		RELEASE indication event
EVE 104		TRANSFER indication event
EVE 106		EXPEDITED indication event
EVE 108		GIVE TOKENS indication event
EVE 109		PLEASE TOKENS indication event
EVE 110A		REJECT indication event
EVE 112		ESTABLISH confirmation event
EVE 116A		RELEASE AFFIRMATIVE confirmation event
EVE 116B		RELEASE NEGATIVE confirmation event

2. Tables de transitions

Table 3. Connection

STATES / EVENTS	STA 1 UNCONNECT	STA 2A wait for ACCEPT message	STA 8 wait for ESTABLISH response
EVE 1 ESTABLISH request	Send CN STA 2A		
EVE 25 ESTABLISH response			Send AC STA 7-13
EVE 10A REJECT request			Send RF STA 1
EVE 11 CONNECT message	EVE 101 ESTABLISH indic. STA 8		
EVE 12 ACCEPT message		EVE 112 ESTABLISH confir. STA 7-13 RC	
EVE 12A REFUSE message		EVE 110A REJECT indication STA 1	

Table 4 . Data Transfer DEF (DT)

STATES / EVENTS	STA 2A wait for ACCEPT message	STA 7-13 CONNECTED
EVE 4 TRANSFER request		DT condit. send DT STA 7-13
EVE 6 EXPEDITED request		send EX STA 7-13
EVE 5 CANCEL request		DT cond. send CL STA 7-13
EVE 15 DATA TRANSFER message		^DT cond EVE 104 Transfer indication STA 7-13
EVE 18 EXPEDITED message	STA 2A ST	EVE 106 EXPEDITED indication STA 7-13

Table 5. Data Transfer DEF (DT),DEF (TR)

STATES / EVENTS	STA 3 wait for DISCONNECT message
EVE 4 TRANSFER request	
EVE 6 EXPEDITED request	
EVE 5 CANCEL request	
EVE 15 DATA TRANSFER message	
EVE 18 EXPEDITED message	EVE 106 EXPEDITED indication STA 3

Table 6. Data Transfer DEF (DT), ^DEF (TR)

STATES / EVENTS	STA 3 wait for DISCONNECT message
EVE 4 TRANSFER request	
EVE 6 EXPEDITED request	
EVE 15 DATA TRANSFER message	^DT cond EVE 104 TRANSFER indication STA 3
EVE 18 EXPEDITED message	EVE 106 EXPEDITED indication STA 3

Table 7. Data Transfer ^DEF (DT)

STATES / EVENTS	STA 2A wait for ACCEPT message	STA 3 wait for DISCONNECT message
EVE 4 TRANSFER request		
EVE 6 EXPEDITED request		
EVE 15 DATA TRANSFER message		EVE 104 TRANSFER indication STA 3
EVE 18 EXPEDITED message	STA 2A ST	EVE 106 EXPEDITED indication STA 3

STATES / EVENTS	STA 7-13 CONNECTED
EVE 4 TRANSFER request	send DT STA 7-13
EVE 6 EXPEDITED request	send EX STA 7-13
EVE 15 DATA TRANSFER message	EVE 104 TRANSFER indication STA 7-13
EVE 18 EXPEDITED message	EVE 106 EXPEDITED indication STA 7-13

Table 8. Disconnection DEF (TR)

STATES / EVENTS	STA 3 wait for DISCONNECT message	STA 7-13 CONNECTED	STA 9 wait for RELEASE response
EVE 3 RELEASE request		DT and TR cond. send FN STA 3	
EVE 26A RELEASE AFFIRMAT response			send DN STA 1
EVE 26B RELEASE NEGATIVE response			send NF STA 7-13
EVE 13 FINISH message		^DT and ^TR cond EVE 103 RELEASE indic. STA 9	
EVE 14 DISCONNECT message	EVE 116A RELEASE AFFIRMAT confirm STA 1		
EVE 33 NOT FINISHED message	EVE 116B RELEASE NEGATIVE confirm STA 7-13		

NOTE D.8

If the Data Token is not defined, ignore the DT and ^DT conditions in this table.

Table 9. Disconnection ^DEF (TR)

STATES / EVENTS	STA 1 UNCONNECTED	STA 3 wait for DISCONNECT message	STA 7-13 CONNECTED	STA 9 wait for RELEASE response
EVE 3 RELEASE request			DT cond send FN STA 3	
EVE 26A RELEASE AFFIRMAT response				send DN STA 1
EVE 26B RELEASE NEGATIVE response				
EVE 13 FINISH message		EVE 103 RELEASE indic. STA 9	^DT cond. EVE 103 RELEASE indic. STA 9	
EVE 14 DISCONN message	STA 1	EVE 116A RELEASE AFFIRMAT confirm STA 1		EVE 116A RELEASE AFFIRM confirm STA 9
EVE 33 NOT FINISHED message				

Table 10 . Abort

STATES / EVENTS	STA 1 UNCONN	STA 2A wait for ACCEPT message	STA 3 wait for DISCONN message	STA 7-13 CONNECTED	STA 8 wait for ESTABL resp.	STA 9 wait for RELEASE response
EVE 2 ABORT request		Send AB STA 1	Send AB STA 1	send AB STA 1	send AB STA 1	send AB STA 1
EVE 31 ABORT message	STA 1	EVE 102 ABORT indic. STA 1	EVE 102 ABORT indic. STA 1	EVE 102 ABORT indic. STA 1	EVE 102 ABORT indic. STA 1	EVE 102 ABORT indic. STA 1

Table 11 . Token Transfer

STATES / EVENTS	STA 3 wait for DISCONN message	STA 7-13 CONNECTED
EVE 9 PLEASE TOKENS request		send PT STA 7-13
EVE 8 GIVE TOKENS request		send GT (see note) STA 7-13
EVE 23 PLEASE TOKEN message	EVE 109 PLEASE T indic. STA 3	EVE 109 PLEASE T indic STA 7-13
EVE 21 GIVE TOKENS message	EVE 108 GIVE T indic. STA 3	EVE 108 GIVE T indic STA 7-13

5.6. Spécifications fonctionnelles des primitives du niveau.

```
1.    S-connect ( init, accp       : sessaddress ;
                  tokenset         : tokenrec ;
                  userdata         : sdutype ;
                  datalg           : integer ;
                  var res, status  : integer ) ;
```

a) spécification

Réalise - l'initialisation d'une demi-session d'adresse ' init ',

- l'envoi d'un message de demande de connexion (CN) vers l'entité-session d'adresse ' accp ',
- et la prise en compte du message de réponse (refus ou acceptation de la connexion).

Le message de demande de connexion contient les paramètres de session définis dans la structure ' tokenset ' :

- choice : spécifie si oui ou non il y utilisation de " jetons ".
- defdt,deftr : booléens spécifiant, lorsque choice est ' vrai ', si oui ou non le data-token et le terminate token, respectivement, sont utilisés.
- dtside, trside : variables comprises entre Ø et 2, spécifiant à quel côté les " jetons " choisis sont assignés :

  Ø : le choix est laissé à l'entité-session recevant le ' connect ' ;

  1 : le côté de l'initiateur du ' connect ' est choisi;

  2 : le côté du receveur du ' connect ' est choisi;

Optionnellement, un message utilisateur de longueur ' datalg ', contenu dans le buffer spécifié par ' data ' pourrait être inclus dans le message de connexion. Cette implémentation l'ignore pour raison de simplicité.

En retour, la procédure garnit ' res ' avec

- une valeur > 0 signalant que la session a été acceptée, et désignant le numéro de connexion-session créée. Ce numéro est à utiliser dans la suite pour référencer cette connexion-session.

- une valeur < 0 signalant un échec :

a) 'tkref' : un au moins des paramètres de session a été refusé par la session distante.

b) 'errparam' : un des paramètres, au moins, est invalide.

c) 'nosessfree': aucune ressource n'a pu être affectée à cette session.

d) 'trspfail' : le niveau transport est hors fonction et la connexion ne peut être réalisée.

e) 'notok' : consulter la valeur de ' status '.

```
2.      S-activate ( init              : sessaddress ;
                     var acp           : sessaddress ;
                     tokenset          : tokenrec ;
                     var res, status   : integer ) ;
```

a) spécification

Réalise l'initialisation passive d'une entité-session, en vue de l'attente sur une adresse-session ' init ' d'une demande de connexion (CN) provenant d'une entité-session distante. Les paramètres de session désirés sont spécifiés dans ' tokenset ' et serviront à la négociation de la session.

Au retour ' res ' sera garni avec

- une valeur > 0 désignant le numéro de connexion-session créée, si la négociation est positive. Ce numéro est à utiliser dans la suite pour référencer cette connexion-session.
- une valeur < 0 en cas d'échec :

a) errparam : un au moins des paramètres est invalide.

b) nosessfree : aucune ressource n'a pu être affectée à cette session.

c) trspfail : le niveau transport est hors fonction.

d) refused : établissement de session refusé pour désaccord sur les paramètres de session.

e) notok : consulter la valeur de ' status '.

Si la connexion est créée, ' acp ' contient l'adresse de l'entité-session ayant lancé la demande.

```
3.      S-data ( sessid            : sessidtype ;
                 userdata          : sdutype ;
                 datalg            : integer ;
                 var res, status   : integer ) ;
```

a) spécification

Réalise l'envoi, sur la connexion-session de numéro ' sessid ' du message de longueur ' datalg ' contenu dans le buffer ' userdata '.

En retour d'appel, la variable ' res ' prend

- une valeur nulle ' ok ' si le message a été correctement envoyé.
- une valeur < 0 en cas d'échec :

a) 'errparam' : un des paramètres, au moins, est invalide.

b) 'trspfail' : le niveau transport est hors fonction.

c) 'nocredit' : autorisation d'émettre non reçue de l'entité-session distante.

d) 'notok ' : modification d'état de la session a eu lieu; examiner ' status ' pour déterminer la raison du non envoi (erreur de protocol , abort reçu...).

Quelle que soit la valeur de ' res ', ' status 'donne des informations sur l'état de la session : réception de PT ou GT, ... voir description de ' status '.

```
4.      S-expedited ( sessid            :  sessidtype ;
                      userdata          :  sdutype ;
                      datalg            :  integer ;
                      ver res, status   :  integer ) ;
```

a) spécification

Réalise l'envoi sur la connexion-session de numéro ' sessid ', d'un message prioritaire, de longueur ' datalg ' < 6 octets, contenu dans le buffer ' userdata '.

En retour d'appel, la variable ' res ' prend

- une valeur nulle ' ok ' si le message a été bien envoyé.
- une valeur < 0 en cas d'échec :

a) 'errparam' : un paramètre, au moins, est invalide.

b) 'trspfail' : le niveau transport est hors fonction.

c) 'notok ' : modification d'état de la session a eu lieu; examiner ' status '.

Quelle que soit la valeur de ' res ', ' status ' donne des informations sur l'état de la session.

```
5.    S-receive ( sessid           : sessidtype ;
                  var userdata     : sdutype ;
                  var datalg       : integer ;
                  var res, status  : integer ) ;
```

a) spécification

Réalise 1) la réception d'un message, sur une connexion-session de numéro ' sessid '. Ce message garnira le buffer ' userdata '; sa longueur sera contenue dans ' datalg '.

2) l'envoi, lorsque le message est reçu, d'une nouvelle autorisation à émettre sur la connexion-session ' sessid '.

En retour, la variable, ' res ' prend

- la valeur 'dtrcvd' si un message de type data est reçu.
- la valeur 'exrcvd' si un message de type expedited est reçu.
- la valeur nodata si aucun message n'est présent.
- une valeur < 0 en cas d'échec :

a) 'errparam' : numéro de connexion-session invalide.

b) 'notok' : modification d'état; consulter 'status'.

' status ' : comme précédemment.

```
6.      S-token-give ( sessid            :  sessidtype ;
                       givedt, givetr    :  bit ;
                       userdata          :  sdutype ;
                       datalg            :  integer ;
                       var res, status   :  integer ) ;
```

a) spécification

Réalise l'envoi, sur la connexion-session de numéro ' sessid ', d'un message de commande signalant à l'entité-session distante que des " jetons " (data : modification du sens de transmission; terminate : droit de terminer une session de façon négociée) lui sont donnés. Les " jetons " donnés sont spécifiés par les paramètres ' givedt ' et ' givetr ', pour le data token et le terminate token respectivement. Ces paramètres ont la valeur 0 si rien n'est donné et 1 pour le " jeton " donné.

Optionnellement un message utilisateur pourrait être inclus au message, mais n'est pas considéré ici. ' datalg ' prend donc la valeur 0.

En retour d'appel, la variable ' res ' prend

- une valeur nulle ' ok ' si le message a été correctement envoyé.
- une valeur < 0 en cas d'échec :

a) 'errparam' : un des paramètres, au moins, est invalide.

b) 'trspfail' : le niveau transport est hors fonction.

c) 'errtoken' : un au moins des " jetons " donnés, est invalide (non défini lors connexion ou aucun spécifié).

d) 'notok' : modification d'état; consulter ' status ' pour information.

' status ' donne des informations sur l'état de la session.

```
7.      S-please ( sessid              : sessidtype ;
                   pleasedt, pleasetr : bit ;
                   userdata           : sdutype ;
                   datalg             : integer ;
                   var res, status    : integer ) ;
```

a) spécification

Réalise l'envoi, sur la connexion-session de numéro ' sessid ', d'un message de commande signalant à l'entité-session distante que l'on requiert pour la session locale des " jetons " appartenant, pour l'instant, à l'entité-session distante. Ces " jetons " demandés sont spécifiés 1) par ' pleasedt ',dans le cas d'une connexion half-duplex, pour demander une modification de sens de transmission (droit d'émettre à son tour);

2) par ' pleasetr ',pour demander d'obtenir le droit d'initialiser une terminaison de session négociée (droit en possession de l'autre entité).

Ces paramètres ont la valeur 0 si rien n'est requis et 1 si le " jeton " est requis.

Optionnellement un message utilisateur contenu dans le buffer ' userdata ' pourrait être inclus au message de commande. Ce cas n'est pas considéré ici.

En retour d'appel la variable ' res ' prend

- une valeur nulle ' ok ' si le message a été envoyé.
- une valeur < 0 en cas d'échec :

a) 'errparam' : un des paramètres, au moins, est invalide.

b) 'errtoken' : un au moins des " jetons " spécifiés (demandés) est invalide (inexistant ou déjà en possession de l'entité-session).

c) 'trspfail' : le niveau transport est hors fonction.

d) 'notok' : modification d'état de la session ; consulter 'status' pour information.

```
8.      S-abort ( sessid            : sessidtype ;
                  userdata          : sdutype ;
                  datalg            : integer ;
                  rs                : rstype ;
                  var res, status   : integer ) ;
```

a) spécification

Réalise l'envoi, sur la connexion-session de numéro ' sessid ', d'une demande de fin de session immédiate. Cette fin de session immédiate peut entraîner la perte de messages pour les 2 entités-session. Sur réception, l'entité-session distante relâche immédiatement la connexion. La raison de l'abort est spécifiée par ' rs ' : (0) non spécifié; (4) erreur de protocole.

Optionnellement un message utilisateur contenu dans buffer spécifié par ' userdata ' pourrait être inclus dans le message ' abort '. Ce cas n'est pas envisagé ici.

En retour, la variable ' res ' prend

- une valeur nulle ' ok ' si le message a été envoyé; la session est alors fermée.
- une valeur < 0 en cas d'échec :

a) 'errparam' : un, au moins, des paramètres est invalide.

b) 'trspfail' : le niveau transport est hors fonction.

c) 'notok' : modification d'état de la session; consulter ' status '; la session est fermée.

' status ' : comme précédemment.

```
9.      S-release ( sessid            : sessidtype ;
                    var res, status   : integer ) ;
```

a) spécification

Réalise l'envoi d'une demande de fin session négociée, sur la connexion-session de numéro ' sessid '. L'autre entité, si le " jeton " terminate est défini, renverra soit un refus de terminer la session, soit un accord pour cette terminaison.

Si le " jeton " terminate est non défini, elle ne peut renvoyer que son accord. Cette méthode assure une fin de session sans pertes de messages. L'entité réceptrice peut en effet compléter ses envois avant de renvoyer l'accord de fin de session.

En retour, la variable ' res ' prend

- une valeur nulle ' ok ' si le message a été transmis.
- une valeur < 0 en cas d'échec :

a) 'errparam' : le numéro de connexion-session est invalide.

b) 'trspfail' : le niveau transport est hors fonction.

c) 'errtrs' : le droit de terminer de façon négociée n'appartient pas à l'entité-session locale.

d) 'notok' : modification d'état; consulter ' status '.

```
IO.     S-not-finished ( sessid          : sessidtype ;
                         userdata        : sdutype ;
                         datalg          : integer ;
                         var res, status : integer ) ;
```

a) spécification

Réalise l'envoi, sur la connexion-session de numéro ' sessid ', du message de refus de terminer la session.

L'appel à cette procédure est conditionnel à l'existence du " jeton " terminate déterminé lors de la connexion, et à la réception préalable d'une demande de fin de connexion.

Optionnellement un message utilisateur pourrait être inclus. Ce cas n'est pas envisagé.

En retour, la variable ' res ' prend

- une valeur nulle ' ok ' si le message est transmis.
- une valeur < 0 en cas d'échec :

a) 'errparam' : un, au moins, des paramètres est erroné.

b) 'notok' : modification d'état ou demande de fin de connexion non reçue; consulter ' status '.

c) 'trspfail' : le niveau transport est hors fonction.

d) 'errnnf' : droit de refus non défini ou droit n'appartenant pas à la session locale.

' status ' : comme précédemment.

```
11.     S-finished-ok ( sessid            : sessidtype ;
                        userdata          : sdutype ;
                        datalg            : integer ;
                        var res, status   : integer ) ;
```

a) spécification

Réalise l'envoi, sur la connexion-session de numéro ' sessid ', du message d'accord de fin de session.

Ce message est une réponse au message de demande de fin de connexion et il clôt la session.

Si le " jeton " terminate est indéfini, l'envoi de ce message est obligatoire pour terminer, sans perte de données, la session en cours.

Optionnellement un message de longueur limitée pourrait être inclus. Ce cas n'est pas considéré ici; datalg doit être nul.

En retour, la variable ' res ' prend

- une valeur nulle ' ok ' si le message a été envoyé. La session est alors fermée.
- une valeur < 0 en cas d'échec :

a) 'errparam' : un au moins des paramètres est erroné.

b) 'trspfail' : le niveau transport est hors fonction; la session est cependant fermée.

c) 'notok' : modification d'état ou non réception d'une demande de fin de connexion; consulter ' status '.

' status ' : comme précédemment.

12. Valeurs de ' status ' :

Abrcvd : message Abort reçu, session terminée (état interne à 'unactivate').

Trspfail : malfonction du niveau Transport détectée, session interrompue et terminée.

Proterr : erreur de protocole détectée, session interrompue et terminée.

DNrcvd : message Disconnect reçu, session terminée si un message Finish n'a pas été reçu précédemment. Auquel cas il reste à envoyer un Disconnect pour terminer définitivement la session.

FNrcvd : message Finish reçu.

NFrcvd : message Not Finished reçu.

GTrcvd : message Give token reçu.

PTrecvd : message Please token reçu.

Notset : valeur neutre.

5.7. Spécifications fonctionnelles des primitives de l'interface Session/Transport.

```
1.      Fromt ( var sessid   :  sessidtype ;
                var msgtype  :  scommand ;
                var buffer   :  sdutype ;
                    cid      :  connidtype ) ;
```

a) spécification

Réalise, sur une connexion-transport identifiée par son numéro ' cid ', le décodage du message se trouvant dans le buffer de réception associé à cette connexion-transport.

Le message, épuré de ses blocs de contrôle, est placé dans le buffer spécifié par ' buffer '.

Le type de ce message est placé dans,'msgtype', et ' sessid ' est garni avec le numéro de connexion-session associé à la connexion-transport concernée.

Les valeurs prises par ' msgtype ' sont :

- UN : message de type inconnu.
- DT : message normal (data message).
- EX : message expédié (expedited message).
- PT : please-token.
- GT : give-token.
- CDT : crédit de donnée session.
- CN : demande de connexion (connect message).
- RF : refus de connexion (refuse message).
- AC : acceptation de connexion (accept message).
- AB : demande de terminaison immédiate (abort message)
- FN : demande de terminaison ordonnée ou négociée (finished message).
- NF : refus de terminer une connexion (non-finished message).
- DN : accord de fin de connexion (disconnect message).

```
2.      Tot ( sessid      : sessidtype ;
              msgtype     : scommand ;
              userdata    : sdutype ;
              datalg      : integer ;
              var result  : integer );
```

a) spécification

Réalise l'envoi (via appel au niveau transport) sur la connexion-session de numéro ' sessid ' d'un message de type ' msgtype 'dont le contenu se trouve dans le buffer spécifié par ' userdata '. Ce message est de longueur spécifiée par ' datalg '.

En retour d'appel, cette procédure garnit ' result ' avec

- 'ok' : si elle s'est terminée correctement.
- 'trspfail' : niveau transport hors fonction; le message n'a pas été envoyé.

Le format du buffer ' buffer 'est, selon les types de message, celui de la fig. 10.

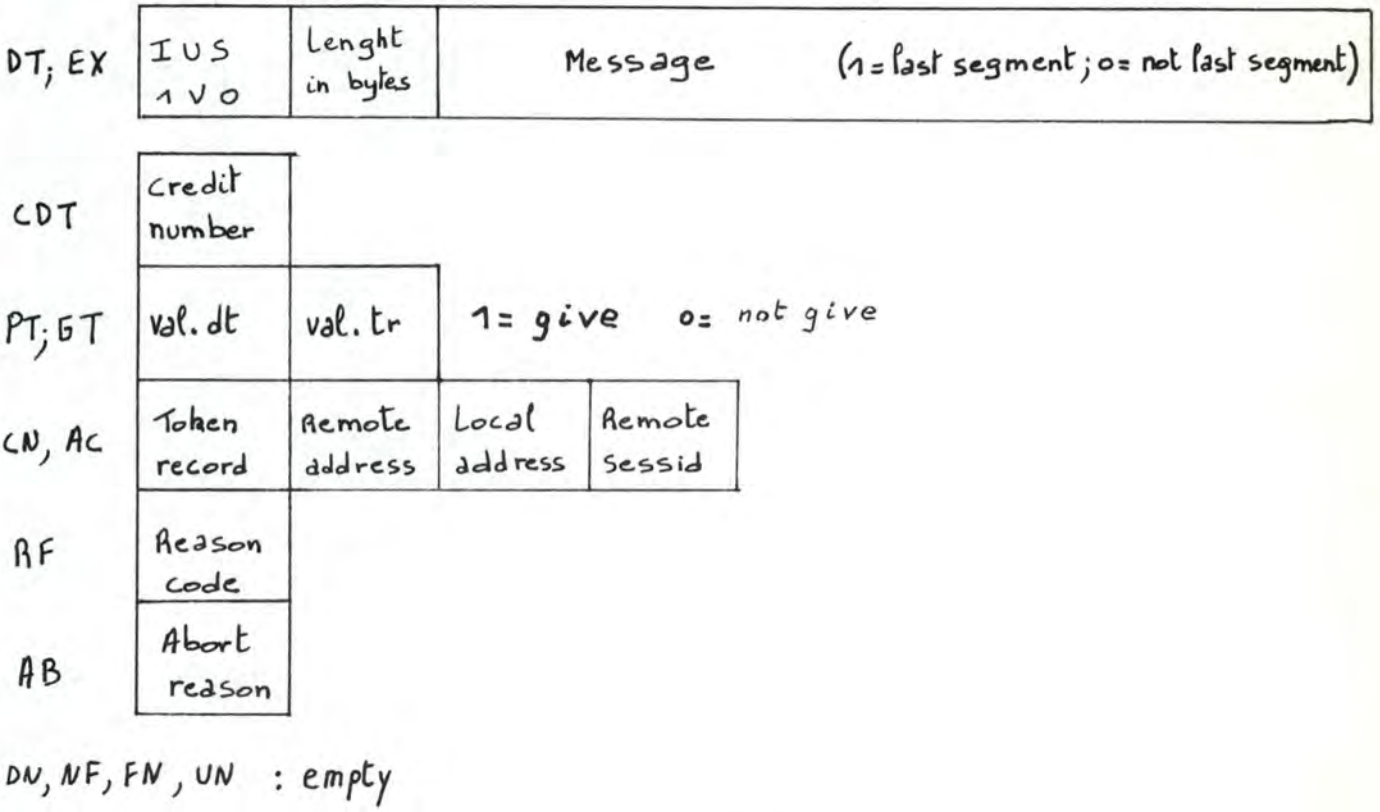


Figure 10: Format des messages

5.8. Spécifications fonctionnelles du moniteur d'interruption Message-Arrival.

Cette procédure assure le traitement des interruptions relatives aux arrivées de messages en provenance du niveau Session.

Elle réalise le décodage de ces messages, leur analyse, les tests de conditions de validité, les modifications d'état résultantes pour les entités-session concernées, ainsi que le réveil de ces dernières (si nécessaire).

6. NIVEAU TRANSPORT.

6.1. Description générale.

Le rôle du protocole de transport et des services de ce niveau, est d'assurer, entre entités-session, un support de communication de bout en bout, transparent et fiable, et ce, indépendamment des moyens de communication sous-jacents (X 25, circuit-switching, datagram, camions, ...).

Les principales caractéristiques requises de ce niveau sont d'après (° ECMA 72) p. 5 :

- la transparence : le service de niveau Transport ne porte aucune restriction quant au contenu, format, codage des informations véhiculées pour le compte de l'utilisateur.

- absence d'erreur : seules les erreurs non récupérables (impossibilité de continuer à maintenir la connexion-transport par exemple) peuvent être répercutées au niveau utilisateur.

- indépendance vis-à-vis des réseaux de communication possibles : le service fourni est homogène et masque les différences sous-jacentes (voir première partie).

- protocole de bout en bout.

- optimisation du coût de communication : en gérant au mieux les ressources de communication disponibles afin d'assurer le service requis par l'utilisateur.

- découplage des adresses-transport et de l'adressage requis par les moyens de communication sous-jacents. Les problèmes de conversion d'adresse logique en adresse physique sont masqués.

6.2. Choix liés aux hypothèses de travail.

Les services et fonctions assurés par ce niveau sont dérivés de l'exemple donné dans (° Tanenbaum 81) p. 328 à 335. Ils ne réalisent qu'un noyau des fonctionnalités présentées dans la description générale.

Nous ne visons pas ici à décrire un niveau Transport complet et réaliste, mais bien une maquette simple de ce qu'il peut offrir à ses utilisateurs (niveau Session).

Les services réseau utilisés sont supposés fiables. Nous posons ici l'hypothèse qu'il s'agit d'un service X 25. Les procédures d'interface traiteront donc les messages comme provenant de, ou étant destinés à des paquets X 25.

6.3. Services choisis.

1. Etablissement et terminaison de connexion-transport : T-listen, T-connect, T-close.

2. Transfert de données : T-send, T-receive.

6.4. Description générale des primitives de service choisies.

1. T-connect : vise à établir une connexion-transport entre deux adresses-transport.

2. T-listen : réalise la mise en attente temporisée (écoute d'une adresse-transport déterminée) d'une éventuelle demande de connexion provenant d'un utilisateur distant (demande résultant d'un appel à S-connect).

3. T-close : termine, sans perte de données, une connexion-transport.

4. T-send : assure la transmission d'un message sur une connexion-transport spécifiée.

5. T-receive : indique le désir de recevoir un message sur une connexion-transport déterminée et la réception de celui-ci.

6.5. Notions liées aux spécifications des primitives.

6.5.1. Contrôle de flux.

Réalisé par un mécanisme de crédit d'émission.

- Une station/entité-transport ne peut envoyer de message (transport Protocol Data Unit) sur une connexion que si elle dispose de crédits pour cette connexion.

- Un crédit est envoyé sur une connexion chaque fois que la primitive T-receive est appelée, en relation avec cette connexion.

- Ce mécanisme assure qu'aucun message n'est envoyé à moins que l'utilisateur récepteur n'ait fait appel à T-receive. Chaque fois qu'un message arrive, on garantit qu'il y a un buffer disponible pour le recevoir.

- La maquette actuelle ne contient pas de buffers multiples pour une connexion (plusieurs crédits d'émission disponibles chez l'émetteur). Les modifications relatives à l'implémentation de ce mécanisme seront discutées dans la section 8.

6.6. Description formelle du protocole de transport.

6.6.1. Etats.

- idle : état neutre, absence de connexion.
- listening : état relatif à la mise en écoute d'une adresse-transport et à l'attente d'une demande de connexion (CallReq).

- waiting : état relatif à l'attente d'une acceptation de connexion (CallAcc) ou d'un refus de connexion (ClearReq).
- queued : état relatif à l'attente d'une " écoute " (T-listen) pour une demande de connexion reçue.
- closing : état relatif à la fermeture en cours d'une connexion-transport.
- receiving : état relatif à l'attente d'un message, venant d'une entité-transport distante, suite à l'envoi d'un crédit d'émission sur la connexion-transport associée.
- sending : état relatif à l'envoi d'un message sur une connexion-transport. Etat stationnaire jusqu'à la fin d'envoi du message.

6.6.2. Evénements.

1. Messages :

- CallReq : requête de connexion-transport.
- CallAcc : acceptation d'une connexion-transport suite à la réception d'un CallReq.
- ClearReq : requête de terminaison de connexion ou, refus de connexion suite à la réception d'un CallReq.
- ClearConf : confirmation de terminaison de connexion suite à la réception d'un ClearReq.
- timeout : fin de temporisation.
- credit : acceptation de recevoir un message.

2. Appels de primitives.

- T-send : envoi d'un message.
- fin de receive : retour à l'état open.
- fin de T-send : retour à l'état open.
- T-listen : écoute d'une adresse-transport.

6.6.3. Diagramme d'états et transitions.

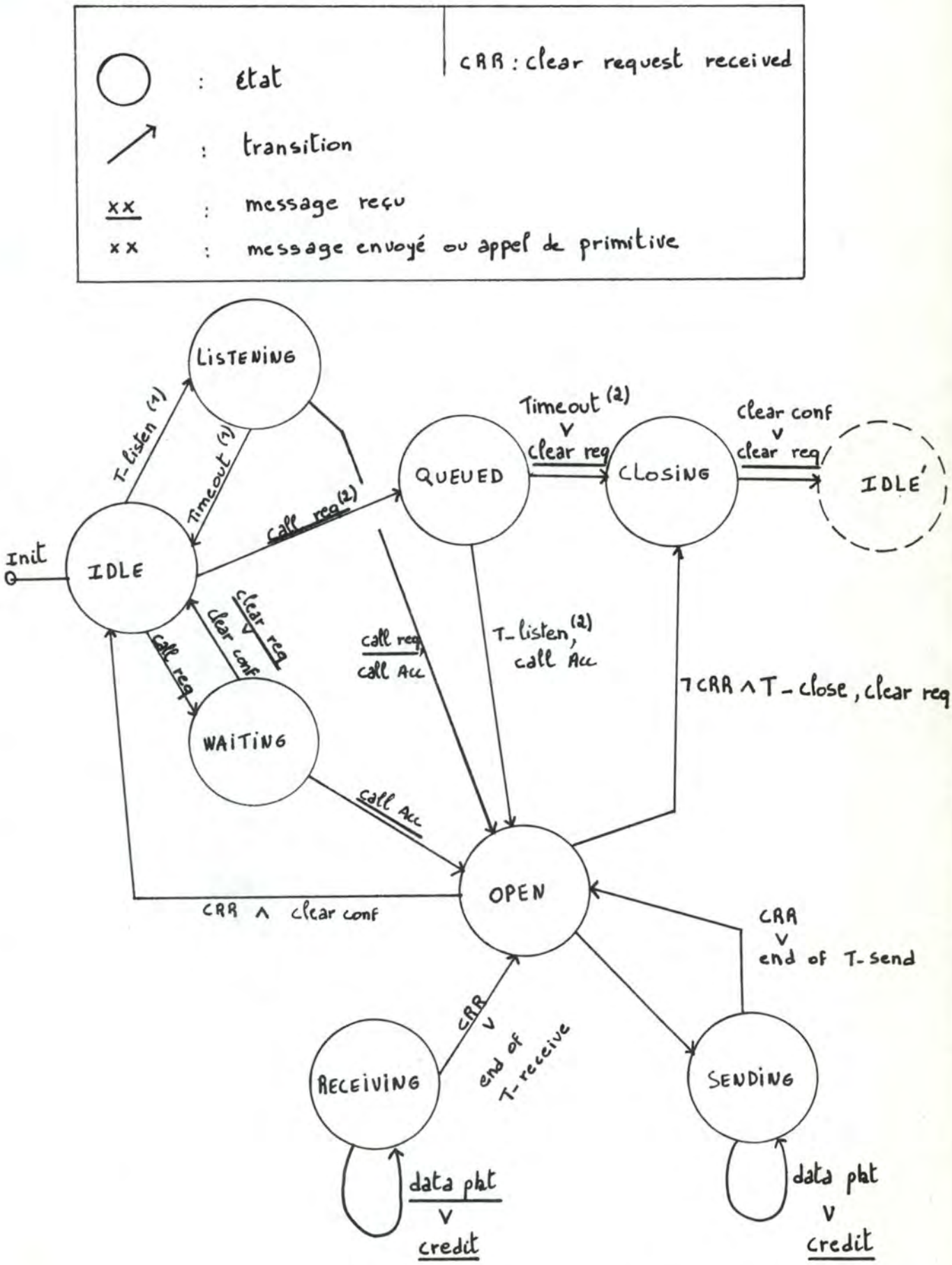


Figure 11 : diagramme d'états et transitions

6.7. Spécifications fonctionnelles des primitives du niveau.

```
1.   T-listen ( t       : transport-address ;
                timeout : integer           ) ; connidorerror;
```

a) spécification

Réalise l'attente passive d'un " call request " (demande de connexion) d'une entité-transport distante. L'écoute est réalisée sur une adresse-transport ' t ', spécifiée en entrée, pour une durée inférieure au délai spécifié par ' timeout '.

La fonction renvoie, en retour :

1) une valeur > 0 si un " call request " a été reçu dans le délai imparti. Cette valeur étant le numéro de la connexion-transport reliant les adresses-transport des deux entités-transport impliquées.

2) une valeur < 0 :

a) 'errtout' : signalant qu'aucun " call request " n'a été reçu dans le délai imparti par ' timeout '.

b) 'errfull' : signalant qu'aucun descripteur de connexion n'a pu être affecté à cette adresse-transport par manque de ressources.

c) 'errparam' : signalant qu'un, au moins, des paramètres d'appel est invalide.

2. T-connect (L, r : transportaddress) : connidorerror :

a) spécification

Réalise la connexion active de deux adresses-transport, appartenant à des entités-transport différentes, en créant une connexion-transport entre ces deux adresses ' L ' et ' r '.

' L ' est l'adresse de l'entité-transport locale, et ' r ' est l'adresse d'une entité-transport distante.

La fonction renvoie, en retour, soit :

1) une valeur > 0 qui est le numéro de la connexion-transport établie entre les deux adresses-transport.

2) une valeur < 0 :

a) 'errejject' : signalant que la demande de connexion a été rejetée par l'entité-transport distante (dû à un Timeout).

b) 'errfull' : signalant qu'aucun descripteur de connexion n'a pu être affecté pour manque de ressources.

c) 'errparam' : signalant qu'un des paramètres, au moins, est invalide.

d) 'errprot' : signalant une erreur de protocole durant l'échange entre les deux entités-transport concernées.

```
3.      T-send ( cid      : connidtype ;
                 bufptr   : msgptr ;
                 bytes    : integer ) : errorcode ;
```

a) spécification

Réalise l'envoi, sur la connexion-transport identifiée ' cid ', d'un message, contenu dans le buffer pointé par ' bufptr ', de longueur ' bytes ' (en nombre de caractères).

La fonction renvoie :

1) la valeur nulle ' ok ' si elle se termine correctement.

2) une valeur < 0 :

a) 'errclosed' : signalant que la connexion-transport est, ou a été fermée par l'entité-transport distante.

b) 'errlong' : signalant une valeur invalide (inférieure à 0 ou supérieure à la taille maximum autorisée ' maxmsg ') du paramètre ' bytes '.

c) 'errprot' : signalant qu'une erreur de protocole a eu lieu dans l'échange entre les 2 entités-transport concernées.

```
4.      T-receive ( cid     :  connidtype ;
                    bufptr  :  msgptr ) : errorcode ;
```

a) spécification

Initialise le buffer pointé par ' bufptr ' pour la réception d'un message et, envoie à l'entité-transport distante un " crédit " l'autorisant à émettre un message (qui ira garnir le buffer apprêté).

' cid ' est le numéro de la connexion-transport concernée; ' bufptr ' est un pointeur vers le buffer de réception.

La fonction renvoie :

1) une valeur nulle ' ok ' si elle se termine correctement.

2) une valeur < 0:

a) 'errclosed' : signalant que la connexion-transport est ou a été fermée par l'entité distante.

b) 'errprot' : signalant une erreur de protocole dans l'échange entre les deux entités-transport concernées.

c) 'errparam' : signalant qu'un, au moins, des paramètres est invalide.

5. T-close (cid : connidtype) : errorcode ;

a) spécification

Réalise la fermeture de la connexion-transport dont le numéro est spécifié par ' cid '.

La fonction renvoie :

1) la valeur nulle 'ok ' si elle se termine correctement.

2) une valeur < 0 en cas d'échec :

a) 'errparam' : le paramètre est invalide (ex : une connexion non ouverte ou déjà fermée).

6.8. Spécifications fonctionnelles des primitives de l'interface Transport/Réseau.

```
1.      Tonet ( cid      :  connidtype ;
                q, m     :  bit ;
                ptype    :  packettype ;
                data     :  packet ;
                count    :  integer ) ;
```

a) spécification

Réalise l'envoi (transmission à la couche réseau) d'un paquet conforme à la norme X 25 sur le réseau.

Le message, envoyé dans le paquet, est contenu dans le buffer spécifié par ' data ' ; sa longueur est ' count '.

Les autres paramètres sont :

- cid : le numéro de connexion-transport associé à un circuit virtuel du réseau.

- q,m : les bits Q et M, du troisième niveau de X 25, qui indiquent respectivement
  - un paquet de données (q = 0) ou de contrôle (q = 1 - le crédit par exemple).
  - que le paquet est un segment (m = 1) d'une chaîne constituant un message, ou, un message complet (m = 0).

- ptype : indique le type de paquet (Call Request, Call Accepted, Clear Request, Clear Confirm, Data, credit).

```
2.      Fromnet  ( var cid      :  connidtype ;
                   var q, m     :  bit ;
                   var pt       :  packettype ;
                   var data     :  packet ;
                   var count    :  integer ) ;
```

a) spécification

Réalise le décodage d'un paquet en provenance du réseau.

Le message contenu dans le paquet est placé dans le buffer spécifié par ' data ', sa longueur est ' count '.

Les paramètres q, m, pt et cid sont les mêmes que ceux décrits dans tonet.

6.9. Spécifications fonctionnelles du moniteur d'interruption Packet-Arrival.

Cette procédure prend en charge le traitement des interruptions relatives aux arrivées de messages en provenance du réseau.

Elle assure leur analyse, la détection d'erreurs (Protocole ...). les modifications d'états des entités-transport concernées, le réveil d'entités endormies, le lancement du moniteur Message-Arrival pour les messages destinés au niveau Session.

6.10. Hiérarchie des primitives du niveau transport de la maquette (fig. 12).

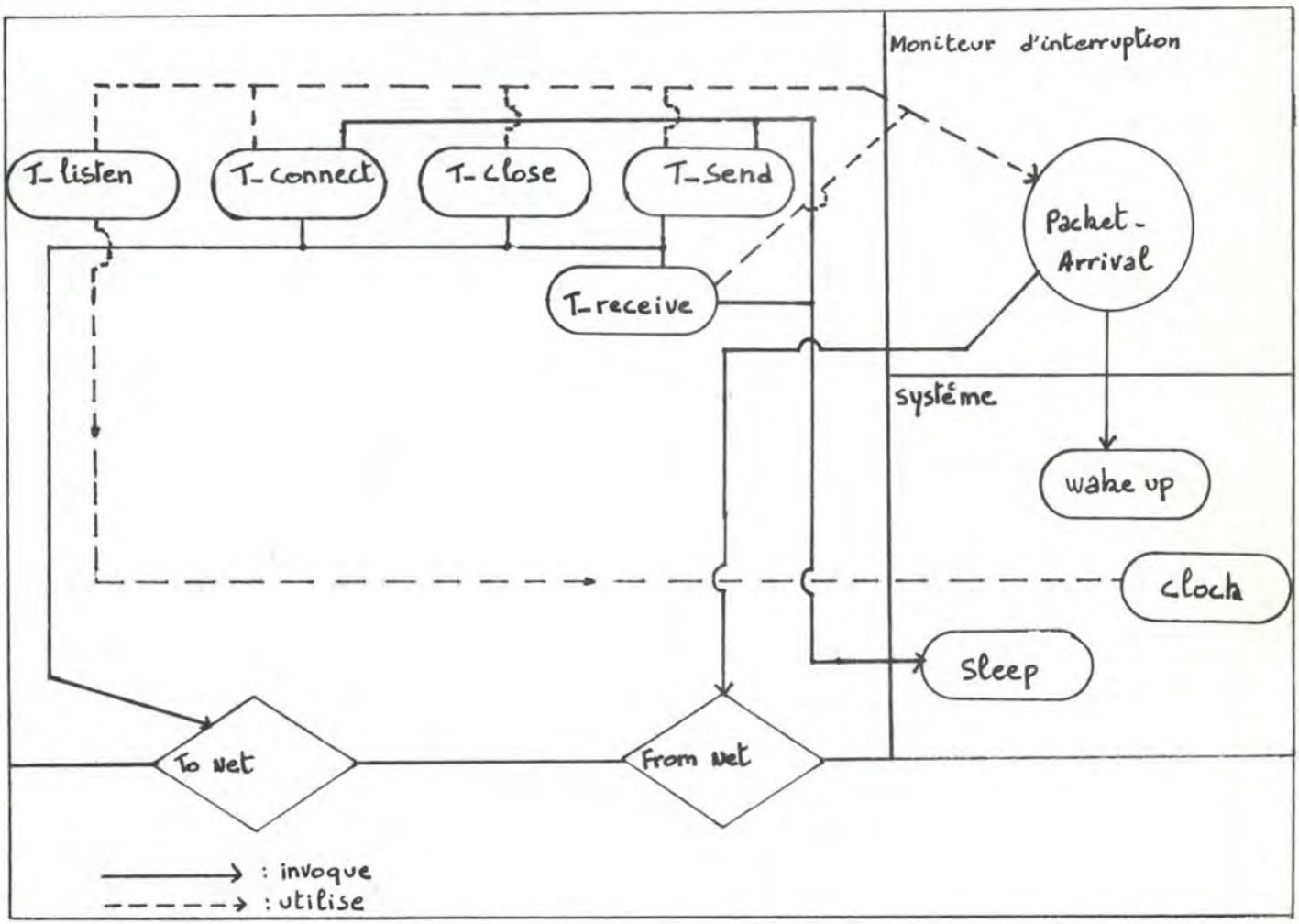


Figure 12

7. PROBLEMES RELATIFS A LA MISE EN OEUVRE.

7.1. Simulation des interruptions.

Elles n'existent pas réellement puisque l'enveloppe de communication constitue, avec son utilisateur, un seul processus.

La procédure SLEEP appelée dans les niveaux Session et Transport simule donc une attente passive de la réalisation d'une condition (fig. 13).

En fait, SLEEP teste si un message, en provenance du réseau, est disponible. Si c'est le cas, SLEEP invoque le moniteur Packet-Arrival qui lui-même invoquera soit Wakeup, soit le moniteur Message-Arrival qui lui, invoquera Wakeup.

Wakeup termine l'exécution de SLEEP et la procédure, ou la fonction, ayant appelé SLEEP, reprend son exécution.

Remarque : le réseau est simulé par un fichier de communication.

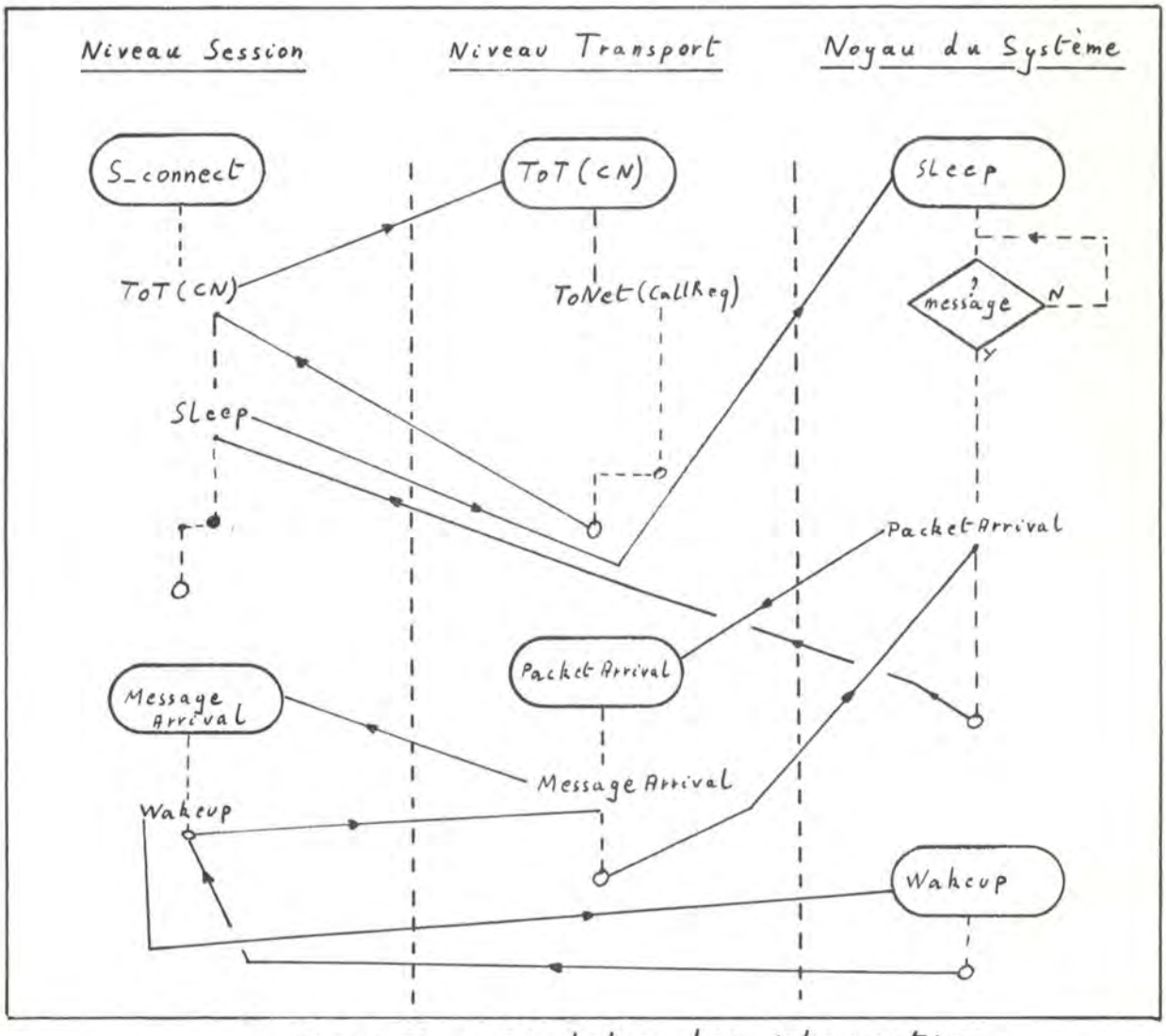


Figure 13 : simulation des interruptions

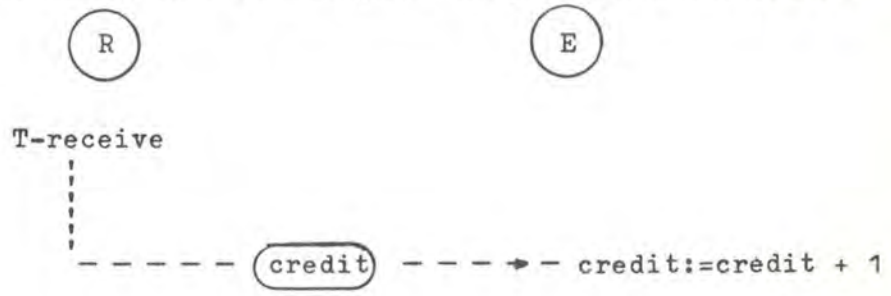
7.2. Flux de messages et commandes au niveau Session.

7.2.1. Problèmes liés à un seul flux pour le niveau Session.

a) en réception, les messages de type commande et de type data (à destination du niveau Présentation) ne seront discriminés que lors de l'analyse du contenu du message.

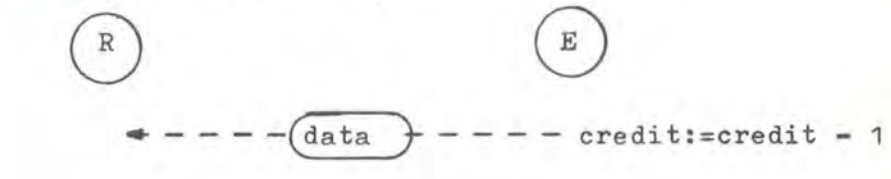
b) Le niveau Session pilote la réception de messages, via le niveau Transport, par l'intermédiaire de la fonction T-receive et du mécanisme de ' crédit ' qui est mis en oeuvre.

c) Par ce fait, une entité session " réceptrice " peut limiter voire bloquer l'envoi de message et/ou de commande chez l'entité-session " émettrice ". Cela nuit à la synchronisation entre ces deux " demi-sessions ". En effet supposons que l'entité-session " réceptrice ", appelée R, signale qu'elle accepte de recevoir un message (donnée ou commande) de E en invoquant la procédure T-receive.



Lorsque la session " émettrice " E désire envoyer un message à R; elle invoque T-send. Le niveau Transport vérifie qu'un crédit d'émission est disponible et envoie le message au niveau Transport servant R.

Supposons que R, ayant pris connaissance du message (données pour niveau P),



ne renouvelle pas son autorisation à recevoir et que de son côté, pour une raison quelconque, E doive terminer la session;

E ne pourra cependant le faire car l'envoi d'une commande ' Abort ' sera bloqué pour absence de crédit d'émission.

d) nous choisissons donc de créer 2 flux logiques entre 2 demi-sessions.

   1) un flux de commandes non régulé.

   2) un flux de données régulé par des crédits de session.

e) il reste cependant, un problème : la non régulation du flux de commandes peut entraîner le recouvrement destructif de certaines données.

7.3. Simulation du niveau Réseau

Le niveau Réseau est simulé par un fichier contenant les paquets échangés.

Afin de réaliser la communication entre les deux processus simulant les deux systèmes distants, il a été nécessaire de mettre en place des utilitaires de dialogue et de synchronisation entre processus (ces fonctions n' étant pas standard sur le Pascal V.U.). Ces primitives ont été conçues et implémentées par monsieur René Verhaegh, que nous remercions pour ce travail, le temps qu' il y a consacré, et pour la disponibilité dont il a fait preuve à notre égard.

Pour une description détaillée, nous prions le lecteur de bien vouloir se référer aux librairies sources (programmes Pascal) ainsi qu' à la librairie de description LIBC(VII), relatives à ces primitives.

8. DISCUSSION DE MODIFICATIONS DE LA MAQUETTE

En raison du manque de temps nécessaire à son développement, ce dernier point ne sera qu' ébauché.

Nous nous limiterons à une énumération, non exhaustive, de modifications possibles en vue d' une amélioration du modèle.

- Création de buffers multiples au niveau Transport (par un système de pointeurs).
- Implémentation des couches par des processus parallèles.
- Multiplicité des utilisateurs du niveau Application.
- Relations N-1 entre une couche (n+1) et une couche (n).
- Multiplexage sur le niveau inférieur.
- Transformation des primitives des différents niveaux en modules d' O.S. au lieu des actuelles routines de librairie.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR,

INSTITUT D´INFORMATIQUE

ETUDE COMPARATIVE D´ ARCHITECTURES DE RESEAUX PAR RAPPORT AU MODELE DE REFERENCE OSI DE L´ ISO

--- A P P E N D I X ---

-- A --

ARCHITECTURE DESCRIPTIONS

Promoteur : Ph. Van Bastelaer.

Meyer Jean-François

Mémoire présenté en vue
de l´obtention du grade de

LICENCIE ET MAITRE EN INFORMATIQUE

ANNEE ACADEMIQUE 1981 - 1982.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR,

INSTITUT D'INFORMATIQUE

--- A P P E N D I X ---

-- A --

ARCHITECTURE DESCRIPTIONS

Promoteur : Ph. Van Bastelaer.

Meyer Jean-François

Mémoire présenté en vue
de l'obtention du grade de

LICENCIE ET MAITRE EN INFORMATIQUE

ANNEE ACADEMIQUE 1981 - 1982.

-- C O N T E N T S O F A P P E N D I X -A- --

CHAPTER 3: ARCHITECTURE DESCRIPTIONS

1. INTRODUCTION 3.1

2. STRUCTURING PRINCIPLES 3.2

3. SNA DESCRIPTION 3.4

3.1. Introduction to SNA

3.2. System cuts 3.6

3.2.1. Network Addressable units (NAUs) 3.7

3.2.2. Network Configuration 3.11

3.2.2.1. Domain :

3.2.2.2. Host :

3.2.2.3. CUCN :

3.2.2.4. Cluster Controller : 3.12

3.2.2.5. Terminal node :

3.2.3. Sessions between NAUs : 3.13

3.2.3.1. Definition :

3.2.3.2. Session types :

3.2.4. End-User (EU) : 3.14

3.2.4.1. definition :

3.2.4.2. Different EUs :

3.2.5. Sessions establishment 3.15

3.2.6. SNA architecture

3.3. Service Cuts 3.16

3.3.1. SNA layering structure

3.3.2. Application layer

3.3.2.1. function :

3.3.3. Function Management Layer

3.3.3.1. Objectives :

3.3.3.2. Functions

3.3.3.3. composition : 3.17

3.3.3.3.1. NAU Service Layer

3.3.3.3.2. Data Flow Control 3.21

3.3.3.4. OSI equivalents : 3.22

3.3.4. Transmission Subsystem 3.24

3.3.4.1. Objectives :

3.3.4.2. Composition :

3.3.4.2.1. Transmission Control Element

3.3.4.2.2. Path Control 3.27

3.3.4.3. Data Link Control 3.32

3.3.4.3.1. Objectives

3.3.4.3.2. Services provided by the DLC :

3.3.4.3.3. OSI equivalent : 3.33

3.3.4.4. Physical Control level

4. DNA description 3.34

4.1. Introduction to DNA

4.2. System cuts

4.2.1. Nodes : 3.35

4.3. Service Cuts 3.38

4.3.1. DNA layering structure

4.3.2. Application Layer

4.3.3. Network Management Layer

4.3.3.1. Objectives

4.3.3.2. Functions 3.39

4.3.3.3. Composition 3.40

4.3.3.4. Services provided to users

4.3.3.5. OSI equivalent

4.3.4. Network Application Layer (N.A.L.) 3.41

4.3.4.1. Objectives

4.3.4.2. Composition

4.3.4.3. Services provided

4.3.4.4. Functions

4.3.4.5. OSI equivalent

4.3.5. Session Control Layer and Network Control Layer 3.42

4.3.5.1. Objectives

4.3.5.2. Functions and composition

4.3.5.3. Services provided to upper layer

4.3.5.4. OSI equivalent

4.3.6. Transport Layer (ISO network level) 3.47

4.3.6.1. Objectives

4.3.6.2. Functions

4.3.6.3. Composition

4.3.6.4. Services provided to upper layer

4.3.6.5. OSI equivalents

4.3.6.6. note

4.3.7. Data Link Control Layer (D.L.L.) 3.51

4.3.7.1. Objectives

4.3.7.2. Composition

4.3.7.3. Functions

4.3.7.4. Services provided to upper layer

4.3.7.5. OSI equivalent

4.3.8. Physical Layer 3.54

4.3.8.1. Objectives

4.3.8.2. Functions

4.4. Protocol cuts 3.55

4.4.1. Introduction

4.4.2. Application layer 3.56

4.4.3. Network Management Layer

4.4.3.1. Composition

4.4.3.2. NICE protocol

4.4.4. Network Application Layer 3.58

4.4.4.1. Data Access Protocol (DAP)

4.4.4.2. Services provided :

4.4.4.3. Dialogue exemple :

4.4.5. Network Service Layer and Session Control Layer 3.59

4.4.5.1. Network Service Protocol (NSP)

4.4.5.2. Services provided.

4.4.5.3. NSP messages types

4.4.5.4. Operation of a logical link and dialogue exemple

4.4.6. Transport Layer (OSI network layer) 3.64

4.4.6.1. General definitions

4.4.6.2. Routing Protocol

4.4.7. Data Link Control 3.67

4.4.7.1. Composition

4.4.7.2. DDCMP :

4.4.7.2.1. Description :

4.4.7.2.2. DDCMP messages types :

4.4.7.2.3. DDCMP operation :

4.4.7.3. MOP (1) :

4.4.8. Physical Layer 3.72

5. DSA description 3.73

5.1. Introduction to DSA

5.2. System Cuts 3.74

5.2.1. Host :

5.2.2. Network Processors : 3.75

5.2.3. Satellites : 3.76

5.2.4. Terminals :

5.3. Service Cuts 3.77

5.3.1. DSA layering structure

5.3.2. Application layer 3.78

5.3.3. Presentation layer 3.78

5.3.3.1. Objectives

5.3.3.2. Services provided

5.3.3.3. OSI equivalents

5.3.4. Session Layer 3.80

5.3.4.1. Objectives

5.3.4.2. Composition and functions

5.3.4.3. Services provided

5.3.4.4. OSI equivalents

5.3.5. Transport Layer 3.83

5.3.5.1. Objectives

5.3.5.2. Functions

5.3.5.3. Services provided

5.3.5.4. OSI equivalents

5.3.6. Network Layer 3.85

5.3.6.1. Objectives

5.3.6.2. Functions

5.3.6.3. Services provided

5.3.6.4. OSI equivalents

5.3.7. Data Link layer 3.88

5.3.7.1. Objectives

5.3.7.2. Functions

5.3.7.3. Services provided

5.3.7.4. OSI equivalents

5.3.8. Physical layer 3.89

5.3.8.1. Objectives

5.3.8.2. Functions and Services provided

5.3.9. Network Administration

5.3.9.1. Objectives

5.3.9.2. Services provided

5.3.9.2.1. NOI :

5.3.9.2.2. NAD :

5.3.9.2.3. NASF :

5.3.9.3. OSI equivalent

5.4. Protocol cuts 3.92

5.4.1. Application layer

5.4.2. Presentation Layer

5.4.2.1. Composition

5.4.2.2. Standard Device Protocol

5.4.2.3. Transparent Protocol

5.4.2.4. Data Description Protocol

5.4.3. Session Layer 3.94

5.4.3.1. Composition

5.4.3.2. Connection Protocol

5.4.3.2.1. Services provided and functions performed

5.4.4. Transport Layer 3.98

5.4.4.1. System Communication Facility

5.4.4.2. Services provided

5.4.4.3. Operation and example of a Transport connection establishment

5.4.5. Network Layer 3.99

5.4.6. Data Link Layer 3.99

5.4.6.1. HDLC Lap-B Protocol

5.4.7. Physical layer 3.99

6. CNA DESCRIPTION 3.101

6.1. Remark :

6.2. Introduction to CNA

6.3. System cuts 3.102

6.4. Service cuts 3.103

6.5. Protocol cuts 3.104

Chapter 3: ARCHITECTURE DESCRIPTIONS

1. INTRODUCTION

Four architectures will be developed in the following. These are SNA, DNA, DSA, CNA.

In a first section, the structuring principles used for those separated descriptions will be briefly exposed.

The four following sections will be devoted to the respective architectures. And in the last part, cross comparisons of those various architectures will be addressed.

- Remark :

  Considering the various sources of information concerning those architectures, the same description degree cannot be reached in what concerns the details.

  This is due, for part, to the reluctentness of some constructors to release technical information, or to the the delays needed to obtain this latter, and for other part to the lake of time needed to cover and complete these descriptions.

  Evenmore, the informations at disposal are not always up to date. We have thus to appologize for that making some points no more relevent.

  The descriptions of CNA and SNA are not complete for the reasons specified above.

2. STRUCTURING PRINCIPLES

The principles used for the descriptions hereafter have been derived from [TUC SP]. This section constitutes a short overview of these latters. For a complete approach, refers to [TUC SP].

The systematic of these Structuring Principles is based on the notion of cut. A cut separates an inworld from an outworld and defines discrete interaction points, at which the inworld and the outworld may interact with each other [TUC SP].

Cuts serve for:

- identifying the units able to interact with each other and their interaction points,
- relating the interaction points of interacting units, and
- defining meaningful interactions at these interaction points.

The cut is the mean to completely abstract from the internal structure and details of the inworld of these units, and enforces to describe the meanings of interactions between these units in terms of external visible behaviour of these units at their interaction points.

The cuts we will use are the System cuts, the Service cuts, and the Protocol cuts.

1. System cuts :

   System cuts serve for achieving a topological decomposition of the real world, or to create mappings between the communication architecture and the real world. They identify individual systems (End systems and Transit systems) as being representatives of those physical components of the real world hosting individual pairwise communication activities (central units, switching devices, transmission lines, ...).

   End systems are systems hosting the communicating Application Entities, i.e. the representatives of communicating parts of the real world.

   Transit systems play the role of being the common media for transmitting these informations between End systems; they perform transmission and switching activities required for the information exchange between communicating End systems.

2. Service cuts :

Service cuts serve for achieving a functional decomposition of individual pairwise communication activities; they identify functional layers.

3. Protocol cuts :

Protocol cuts serve to coordinate system cuts and service cuts with respect to individual pairwise communication activities; they identify protocol entities.

The purpose of Service and Protocol cuts is to define a virtual structure for systems, thus determining their communication behaviour. The purpose of this virtual structure is to describe the structure of communications they must be able to maintain.

3. SNA DESCRIPTION

3.1. Introduction to SNA

SNA defines a unified set of commands, procedures, message formats and protocols used to facilitate data communication between SNA compatible products [IBM C].

SNA [Cypser 78], is a network architecture intended to allow IBM customers to construct their own private networks, both hosts and subnet.

Prior to SNA, IBM had several hundred communication products, using three dozen teleprocessing access methods, with more than a dozen data link protocols alone. The basic idea behind SNA was to eliminate this chaos and to provide a coherent framework for loosely coupled distributed processing.

Given the desire of many IBM's customers to maintain compatibility with all these (mutually incompatible) programs and protocols, the SNA architecture is more complicated in places than it might have been had these constraints not been present.

SNA also performs a large number of functions not found in other networks, which, although valuable for certain applications, tends to add to the overall complexity of the architecture.

SNA has evolved considerably over the years, and is still evolving.

A SNA network consists of a collection of machines called Nodes, of which there are four types approximately characterized as follows.

- Type 1 nodes are terminals.
- Type 2 nodes are controllers, machines that supervise the behaviour of terminals and other peripherals.
- Type 4 nodes are Front End processors, devices relieving the main CPU of communication work.
- Type 5 nodes are the main hosts themselves. (some controllers have some host-like properties in reason of distributed processing).

Each node contains one or more Network Addressable Units (NAUs). A NAU is a piece of software that allows a process to use the network. It is an entry point into the network for user processes.

3.2. System cuts

SNA appears to be a centralised controlled, hierarchical network.

We can identify four physical components :

1. Host nodes :

   Equivalent to a CPU with Operating System, Access Methods,and a Data Base. It is responsible for Data processing, D.B. processing and communication system network management.

2. Communication Controller Nodes (CUCN) :

   Responsible for many Communication System (CS) functions as control of the remote network, that is attached to it; acting as a slave of the host node for which it carries out instructions and messages.

   it is also responsible for

   - control of the communication lines, deleting and inserting characters.
   - code translation.
   - activation/deactivation of lines.
   - error recovery.

   It can act as a F.E.P. to the H.N. (Host Node).

3. Cluster controller Nodes (CCN) :

   - provide remote locations with access to the D.B. or services at the H.N.
   - consist of programmed controllers supporting devices and containing data and processing storage.

- process data and act as stand alone systems servicing their terminals.
- linked to the CUCN by SDLC lines.
- exemple : a 3600 firmware communication system

4. Terminal Nodes (TN) :

- send and receive data from/to the H.N.
- support the attachement of some devices.
- linked to the CUCN by SDLC lines.
- not programmed.

End Systems are H.N., CCN and TN .

Transit systems are CUCNs.

3.2.1. Network Addressable units (NAUs) (fig. 12):

In each of these cuts we have to identify units managing the overall communications for the purpose of end users.

Those coded units, directly addressable, are called Network Addressable Units (NAUs).

- The NAU is a resource managed by the communication system. It provides for end users access to the C.S., and provides services for communication management. [IBM : General Informations]
- NAUs are the origins and destinations of information units flowing in the C.S. .
- Each NAU has a Network Name by which end users identify it . A Network Name is not unique for all its end users but the correlation of names has

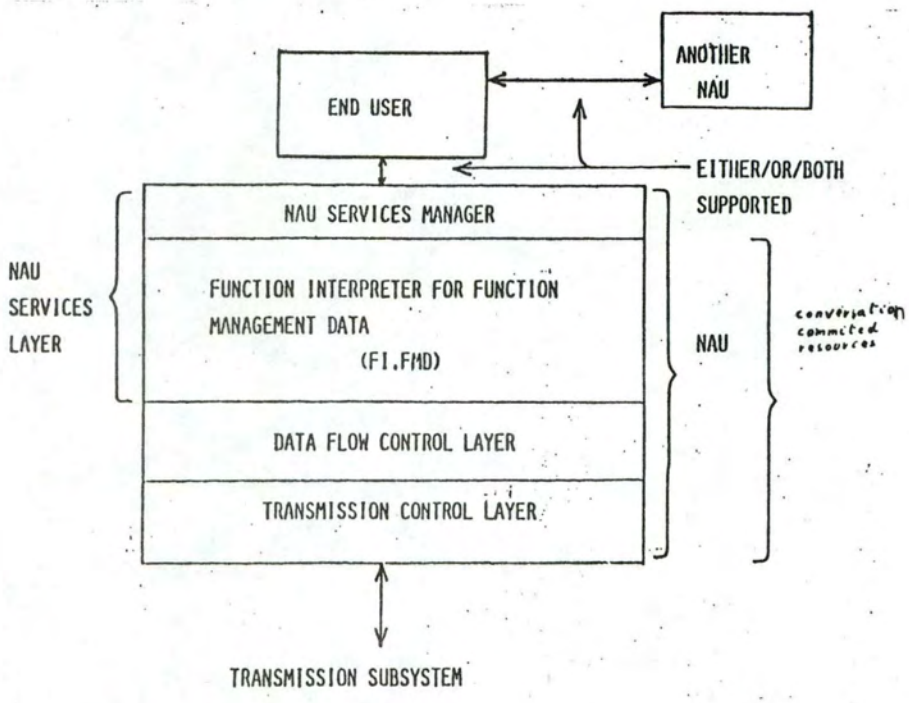


Figure 12: THE NAU STRUCTURE
END-TO-END SERVICES

to be known by the network.

- each NAU has a Network Address assigned by the C.S.. This address is used in the Transmission Subsystem and uniquely identifies the location of the NAU in the C.S. .

- NAU's interact with each others through a set of functions,called a Session, which defines a logical connection between the NAUs implicated.
  One session exists per pair of NAUs but a NAU can support multiple sessions,each of them with an apparied distant NAU.
  In a session each NAU implicated is refered to as a Half Session.

- A given node can host several NAUs.

SNA defines three types of NAUs (fig. 13):

1. System Service Control Point (SSCP) (fig. 14):

   a) definition :

   a special purpose NAU, located in the Host Node, used for network management. (there is no SSCP in other nodes.)
   An SNA network may have one or more SSCPs, each of which managing a portion of the network called a Domain.

   The SSCP has complete knowledge of, and control over, all the front ends, controllers,and terminals attached to the host, pieces of hardware and software constituting the domain.

   b) functions :

   - general management of a control domain and its resources .
   - bringing up and shutting down the network.
   - establishing logical connections (sessions) between the other NAUs.
   - recovering in case of contact failure between components.
   - providing interface to the network operator services for the domain it controls.

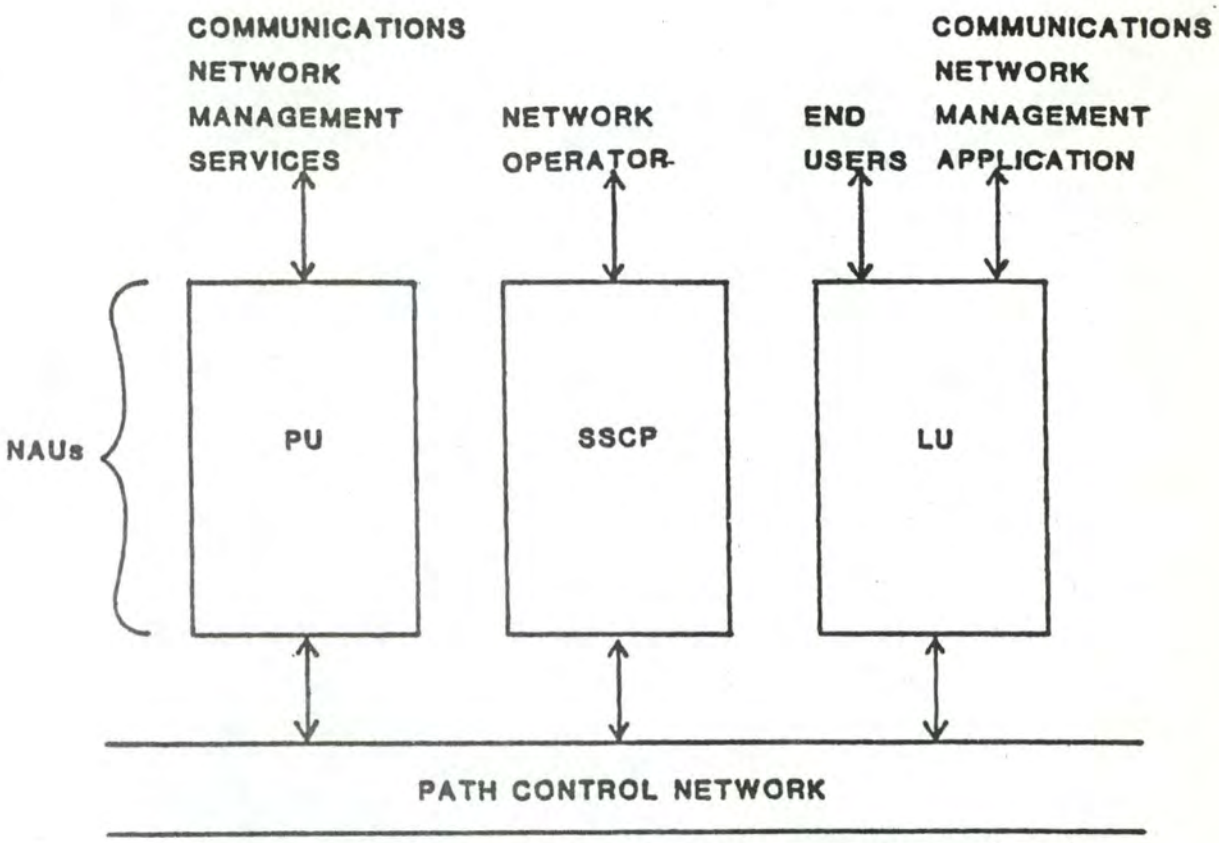


Figure 13 . Types of Network Addressable Units (NAUs)

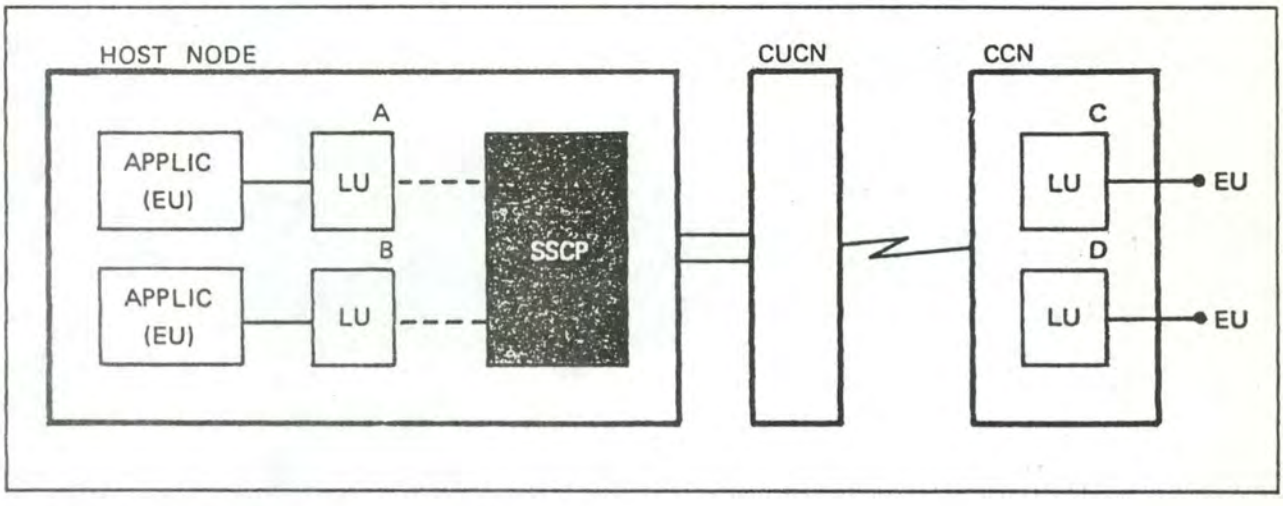


Figure 14: System Services Control Point (SSCP)

- managing and establishing session with other domains.

2. Physical Units (PU) :

a) definition :

A PU is located either in a CUCN, a CCN or a TN. It consists of programs providing physical services to the node for the communication management.
Each node that has been defined to an SSCP has at least one PU.

b) functions :

- SSCP and PU together control the network cofiguration and the data transportation resources provided by the nodes in the domain of the SSCP (this link between SSCP and PU is achieved by a session establishment which is part of the bringing up process.).
- activates/deactivates communicaton links.
- perform services for the SSCP.

The three following types of PUs are identified :

a) CUCN PU: dedicated to the control of its sphere of programing and hardware.

b) CCN PU : supports physical attachments ....

c) TN PU : supports terminal communication functions.

3. Logical unit (LU) :

a) definition :

a LU is located in HN, CCN or TN.

It is a window or port through which the end-user accesses

- the SSCP provided services to help in establishing and supporting logical connections between LUs,and
- the resources addressable by the SSCP.

b) Functions :

- provides ports for end users.
- supports communication between end users by editing or transforming correlating requests/responses from the end user.
- runs data flow procedures to control information flow between end users (EU).
- supports at least two session types : one with the SSCP, one with another LU.
- the number of LUs in a node is implementation dependent.
- LUs in different nodes may have various dedicated functions and provides differents services.

c) LU types :

LUs in different nodes may have various dedicated functions and provides different services.

The three following LUs are identified :

1) HN LU : which can be thought of as programming and control blocks. This programming includes a C.S. control program that can send/receive information to/from remote node for EU.

2) CCN LU : similar to the HN LU.

3) TN LU : hardware logic, has less capabilities than the others.

3.2.2. Network Configuration

SNA network is constituted of multiple interconnected domains or of one single domain.

3.2.2.1. Domain :

A domain is the collection of network resources controlled by the SSCP located in the host managing this domain.
An SNA network domain is thus constituted of Hosts, CUCNs, CCNs, TNs interrelating together.

In a domain each resourses/devices has its address known by the SSCP in the Host.

3.2.2.2. Host :

It houses one SSCP, PU and LUs and controls, with the SSCP, a domain of CUCNs, CCNs, TNs and subnetwork.

It may be in relation with other Hosts through the intermediary of its CUCN acting as a Front End Processor (FEP) for communications (called SSCPs Sessions).

The Host is connected to the CUCN (single or multiple) by a channel connection.

3.2.2.3. CUCN (type 4 unit) :

- handles transmission service for a subarea of the network domain.
- controls communication lines.
- linked to the other CUCNs by switched links.
- controls the communication pathes for its subarea and provides transmission services to other nodes and subareas.
  It does not normally contain LUs but act as a pipe for transmiting information between LUs, PUs, SSCPs, located in other nodes.

Subarea : group of NAUs sharing a common subarea address (fig. 15) but having their own element address.

Subarea address is handled by a CUCN which routes the informations to/from the nodes of this subarea using their element address.

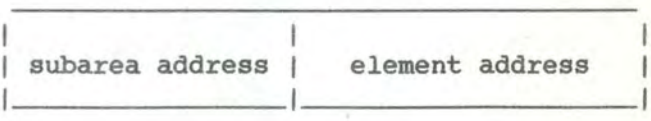


figure 15 : NAU 's adress

3.2.2.4. Cluster Controller (type 2 PU) :

- has no subarea responsabilities.
- controls LUs supporting applications and devices.
- it requires a CUCN to complete the network service it offers to the LUs (network address transformation to local form and reverse; ...).
- can have up to 256 NAUs and one PU.

3.2.2.5. Terminal node (type 1 PU) :

- controls LUs supporting devices
- less powerful than CCNs
- requires a CUCN to complete network services

The fig. 16 illustrates a one domain network containing three subareas.

Fig. 17 and 18 show configurations of multidomains networks. The former shows connected nodes and the latter, the various nodes (cuts) with their logical and control units and, the cross-domain relations.

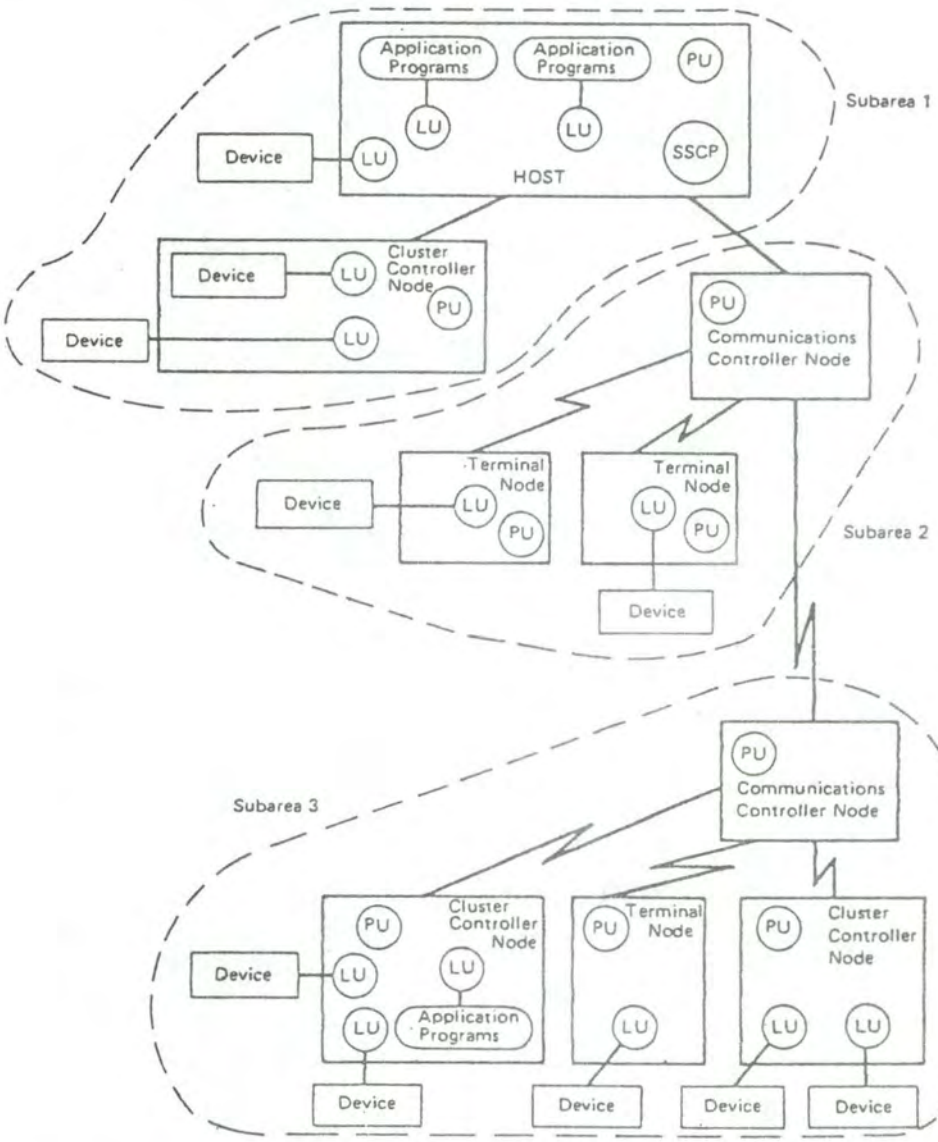


Figure 16: Three subareas within one domain.

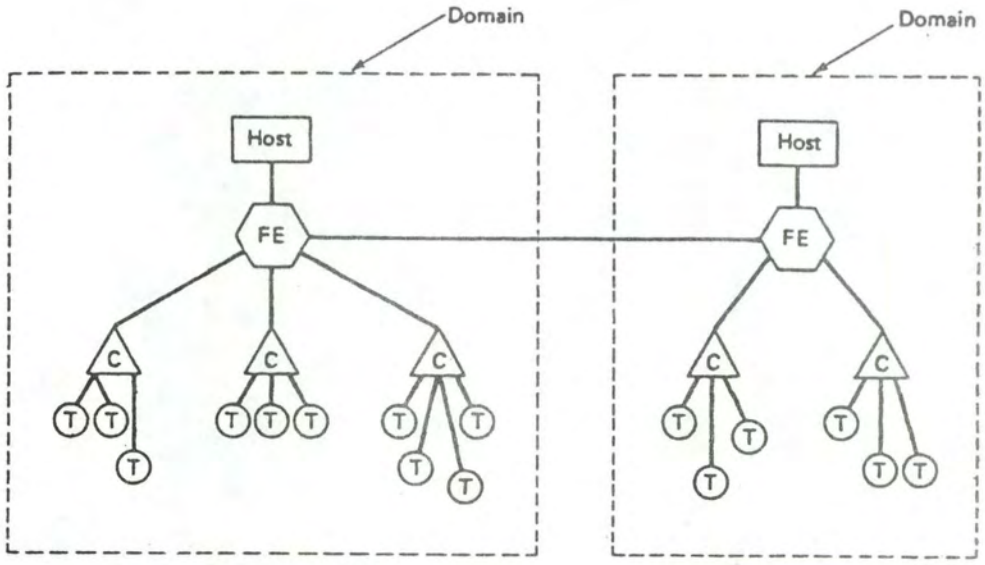


Figure 17: A two domain SNA network. FE = Front End, C = Controller, T = Terminal.

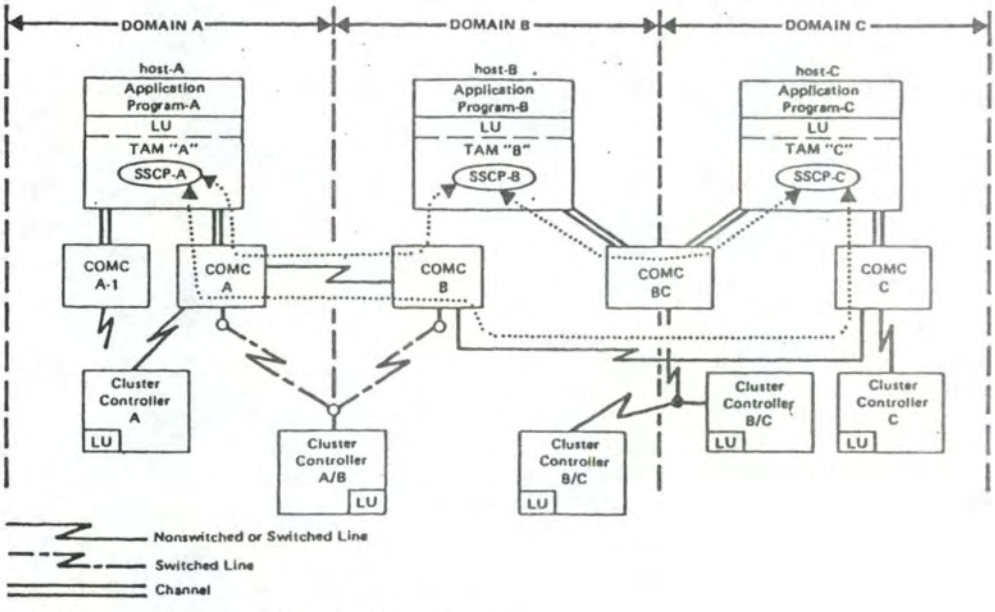


Fig. 18: Illustrative configuration of a multidomain network.

3.2.3. Sessions between NAUs :

3.2.3.1. Definition :

> The Session is a temporary logical connection between NAUs for exchange of messages in accordance with ground rules (flow pacing, recovery facilities, waiting for responses, grouping of responses/requests, data formatting, ...) [SNA].

Architecturally, the session is a set of functions that are used to support the interaction between two NAU service managers. The session is composed of two half-sessions, each being user-oriented functions at one end of the interaction.

The pair of half-sessions provide end-to-end services tailored to serve end-users or NAU S.M. (Service Manager).

"The two ends of a session are not symetric. One ends is designated as the primary and the other as the secondary. The primary usually has more power and responsability than the secondary.
Remember that in the original SNA release there was only one host in the entire network, so the host was the primary and the terminal the secondary. The same assymetry is present in SDLC " [Tanenbaum 81].

3.2.3.2. Session types :

1) SSCP-SSCP : for central Host to central Host connections, and cross-domain Sessions.

2) SSCP-PU :

for node activation by the SSCP,
for control of the physical configuration,
for control of the individual nodes and their resources,
for link activation.

3) SSCP-LU : for requesting LU-LU session establishment and termination.

4) PU-PU : for network control purpose.

5) LU-LU : for data flow between end-users (EUs).

3.2.4. End-User (EU) :

3.2.4.1. definition :

> The ultimate source or destination of information flowing through SNA system. An EU may be an application program, an operator, or a data-medium. EU are located at the H.N., CCN, TN.

EUs appear to be the entities wishing to communicate; the consummers and suppliers of information. They are not part of the network system.

Communication between EUs is achieved through the intermediary of LUs and LU-LU sessions which provide the EUs with various services.

The whole network system aims to serve as media-mean between these Users or 'persons' and, may be viewed as the set of system cuts.

3.2.4.2. Different EUs :

1. The most general part of EUs is a program. It may be a simple application program or a complex one . The program EU may also interface to other I/O devices not visible to the SNA network.
2. Operator EU : the second possible form of EU interface to SNA is a human operator.
3. The third possible form is storage medium that may operate directly via a NAU and SNA protocols. [Cypser 1978]

3.2.5. Sessions establishment

Any LU wishing to communicate with another LU has to request this connection to its own Host's SSCP. The request may be relayed by CCNs,CUCNs inside the domain of the SSCP.

This latter either provides the session if the other LU is in its control domain, or initiates a session with another SSCP controlling the domain housing the other LU.
It will then pass the request to this SSCP which will initiate the session with the LU concerned.

The data exchange between LUs can then start (if both agree on); it doesn't pass anymore through the two SSCPs but through the CUCNs achieving the physical path.

example : LU 1, on fig. 18, wishing to talk to LU 2

1. request of LU 1 goes to SSCP-A following the path CCN-A, CUCN-A, SSCP-A.
2. SSCP-A initiates a session with SSCP-C through CUCN-A, CUCN-B, CUCN-C, SSCP-C. A LU 1-SSCP-C session is then established.
3. SSCP-C recognizes LU 2 and initiates first a SSCP-C-LU 2 sesion through CUCN-C, CCN-C.
4. SSCP-C finally establishes the LU 1-LU 2 session (if possible). The data flow path bypasses the different SSCPs and goes through CCN-A, CUCN-A, CUCN-B, CUCN-C, CCN-C.

3.2.6. SNA architecture

The general architecture of SNA can be viewed as a hierachical relationships inside the control domain of host's SSCP central computers and horizontal/non hierachical ones between host's SSCPs themselves.

3.3. Service Cuts

3.3.1. SNA layering structure

The layering decomposition of SNA is that shown in fig. 19.

3.3.2. Application layer

3.3.2.1. function :

User's application processing.

3.3.3. Function Management Layer

3.3.3.1. Objectives :

Presentation of information from one application to the other.

3.3.3.2. Functions

- Initiate connection requests between LUs.
- Accepts data from the Application Layer (AL) and presents it to the Transmission Subsystem Layer (TSL) for transmission.
- Accepts data from TSL and presents it to AL .
- Controls data flow.
- Provides any devices dependencies.
  [SNA self study course -ch 2-3]

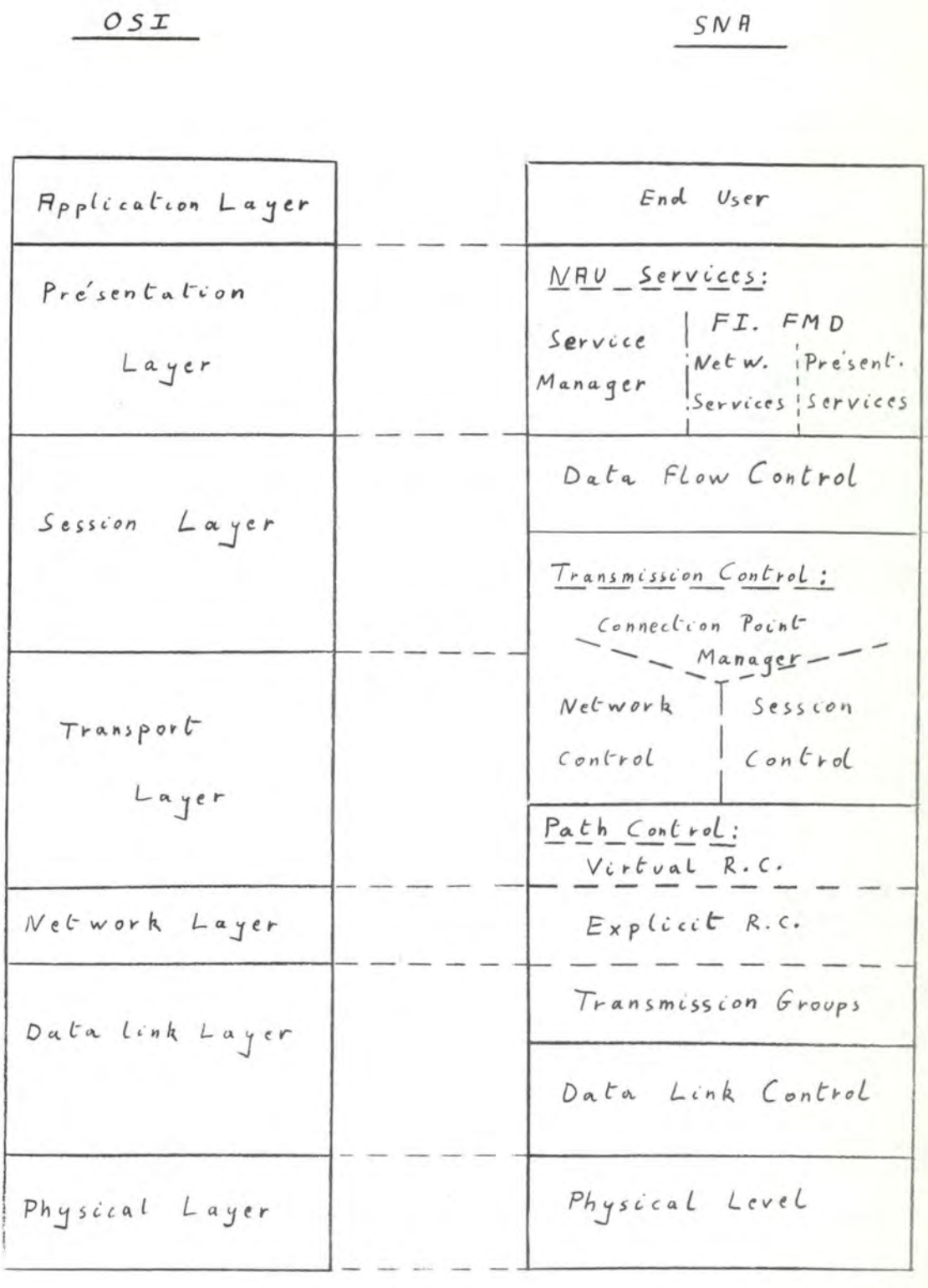


Figure 19: SNA Layering decomposition versus OSI

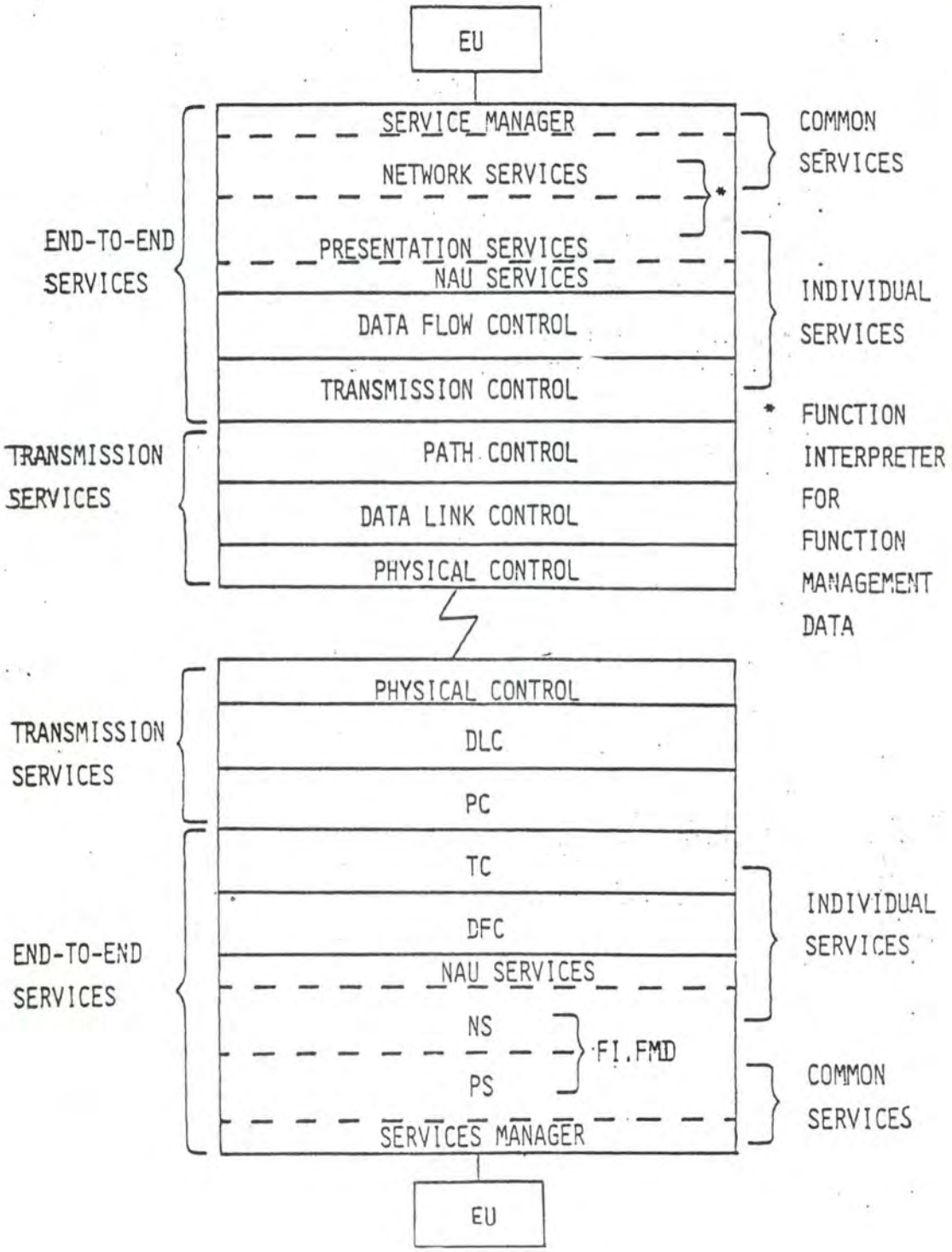


Figure 196 : SNA Layering decomposition

3.3.3.3. Composition :

This layer is divided in two sublayers :

1) NAU Service Layer

2) Data FLow Control

3.3.3.3.1. NAU Service Layer

NAU service layer consists of three parts :

1. NAU Service Manager

2. Network Services

3. Presentation Services

The two last being part of the Function Interpreter for Function Management Data (FI.FMD).

1. NAU Service Manager

Provides end user coordination with the session it is communicating with, and services like session activation/deactivation.

2. Network Services

Functions that provide physical and logical resources management including session establishment and configuration management, plus services needed to control and maintain the network, the node, the session.

We can identify common services (needed for all NAU-NAU sessions) and end-to-end services (needed for a particular session).

These services are distributed differently between the types of NAUs accordingly to the aim they have

to achieve.

a) SSCP network services :

Functions that serve other NAUs in a control domain in order to control network configuration, session establishment/termination, and, manage the control domain.

b) LU network services :

- Concerned with logical connections such as session establishment/termination in collaboration with the SSCP.
- Present in all LU.
- Can call SSCP N.S. for assistance on request from a EU.

c) PU network services :

- Concerned with configuration resource management.
- Works in collaboration with the SSCP N.S. for functions such as bringing up, shutdown, reconfiguration, tracing, testing, ...

d) Functional description of network services

We can define 5 categories of N.S. and their repartition (fig. 20):

	SSCP NS	PU NS	LU NS
configuration S.	*	*	
Measurement S.	*	*	*
Maintenance S.	*	*	*
Session S.	*	*	*
Network operator S.	*		

figure 20 : network service repartition [Cypser 78].

Configuration services : supported on SSCP-PU sessions.

- Used to initially configure the network at start-up time (using tables),
- to modify it,restart it, shut it down,
- to activate/deactivate links, PUs, LUs ,
- to control physical configuration,
- to modify path control routine tables.

Measurement services : supported on SSCP-PU/LU sessions.

Used to measure the use of certain network resources.

These services are not yet precisely defined.

Maintenance services : supported on SSCP-LU/PU sessions.

Used to perform testing and tracing of network facilities

Session services : supported on SSCP-SSCP/LU/PU sessions.

Used to assist LU in activating LU-LU sessions (ground rules establishment,..)

- for resolution of network names
- checking of passwords,
- allocation of simultaneous LU sessions.

It is located partly in SSCP S.S. and LU S.S.

Network operator services :

Used for operator communications with SSCP, to access configuration, maintenance and session services, and to optimize network

operations.

By this way, the operator is able to

- shut down and/or start up the network;
- to control trace of activity;
- to activate links, and initiate all the services provided by Configuration services, Management and Session services.

3. Presentation Services (P.S.) :

Presentation services are located in the LUs, at both end of every session, but predominantly at one end, the host, achieving the most part of the P.S. for a terminal cluster, a terminal node,... .

They provide data transformation and formatting support for EUs such as programs, printers, VDS, ..., and accomodate the data transfered to the EU requirements.

Presentation services are paired, one in each half-session/LU.

" A pair of Presentation services change the view of the information so as to better match the needs or the language of each end user or NAU service Manager." [Cypser 1978]

Differents sets or classes of P.S. exist that are selectable by the EU by use of request commands.

3.3.3.3.2. Data Flow Control

The Data Flow control has the function of accomodating the particularities of messages direction and intermittency demanded by EU (Duplex, Half-Duplex Flip-Flop, Half-Dulpex Contention send/receive modes).
It also provides dialog control and correlation of requests and responses. [NCR]

It lies between the FI.FMD and TC elements, is end user oriented.

It maintains the integrity and order in data flows exchanged by NAUs (or half-sessions), policies the adherence to DFC agreements, reports errors, help to recovery. [Cypser 1978]

Each session contains a DFC element tailored to that session.

Function : this layer aims to maintain the integrity of the data (message) flow by responding to error conditions and confirming receipt of messages ,and, to maintain data flow order.

To do that, it achieves four types of actions or services :

1) passes, intact, the message received from either NAU services or T.C. services to either under or upper layer services;

2) generates or passes along the correct indicators for each messages, in order to determine how is the message to be handled,

3) policies Data Flow rules and generates error indicators in case of protocol violation;

4) generates its own control messages when requested to do so by EU or NAU services.

3.3.3.4. OSI equivalents :

The corresponding OSI layer and functions are

1. For the Data Flow Control :

   Session layer:

   - Dialog management.
   - Quarantine function.
   - Recovery function.
   - Expedited data exchange.
   - Session connection synchronization.

   Transport layer:

   - End-to-end segmenting/ blocking.
   - Interaction management.

2. For the Presentation Services :

   Presentation layer.

   - Image negotiation.
   - Data format and transformation management.
   - Encryption.
   - Compaction.

3. For the Network layer :

   Application layer :

   - Network resources management.
   - Network monitoring (error detection/recovery, reconfiguration, reports on operation).
   - Physical connection management.

   Presentation layer :

- Part of Image negotiation.

4. NAU Service Manager :

Session layer :

- A part of the establishment/termination services.

3.3.4. Transmission Subsystem

3.3.4.1. Objectives :

Transparent routing and movements of data units between origins and destinations.

3.3.4.2. Composition :

1. Transmission Control Element
2. Path Control Element
3. Data Link Control Element

3.3.4.2.1. Transmission Control Element

Transmission Control is the lower layer of each half-session (NAU), located between DFC and Path Control (PC).

It provides its users (NAUs) with direct access to the Transmission Subsystem, establishes sessions between NAUs.
It also control flow rate between NAUs and maintains messages order.

One instance of TC called TCE(Transmission Control Element) exists for each session and is composed of three components dedicated to sessions. [Cypser 1978]

1. Connection Point Manager (CPM) :

   Coordinator of all the flows for one Half Session.

   Interface for the Half Session to the Transport Subsystem.

   It performs most of the transport control jobs once the session is established

(routing, sequencing, pacing, ...).

- Services provided by the CPM

1) Routing of incoming Request Units (RU), from the transmission network, to either Session Control, Network Control, DFC or FI.FMD . (this routing is done in accordance with indications contained in the Header of the RU.)
Merging of RU emanating from the four previous 'units' into one flow to the PC layer.

2) To construct the Request Header (RH) for all RUs emanating from the NAU (originated in DFC or FI.FMD, TC functions, service control or Network Control), and the control parameters associated (fig. 21).

To send the RH-RU combination (BIU) to the Path Control.

3) Generation of sequence numbers or identifiers for each messages leaving the NAU.
Checking of input flow sequence numbers and detection/indication of out-of-sequence incoming messages.
Coordinating responses with requests and keeping them in proper order.

4) Pacing (flow rate control) by peer Protocols which assume that messages are sent/received at a rate the concerned CPM can handle. And that in accord to ground-rules agreed on at session establishment.
This corresponds to the OSI transport layer flow control.

2. Session control components :

Front office type of functions that acts as a control point and coordinate work of others components.

It is used to establish a session and to obtain resources required for a session. It keeps track of session status and provides supports for starting, clearing,

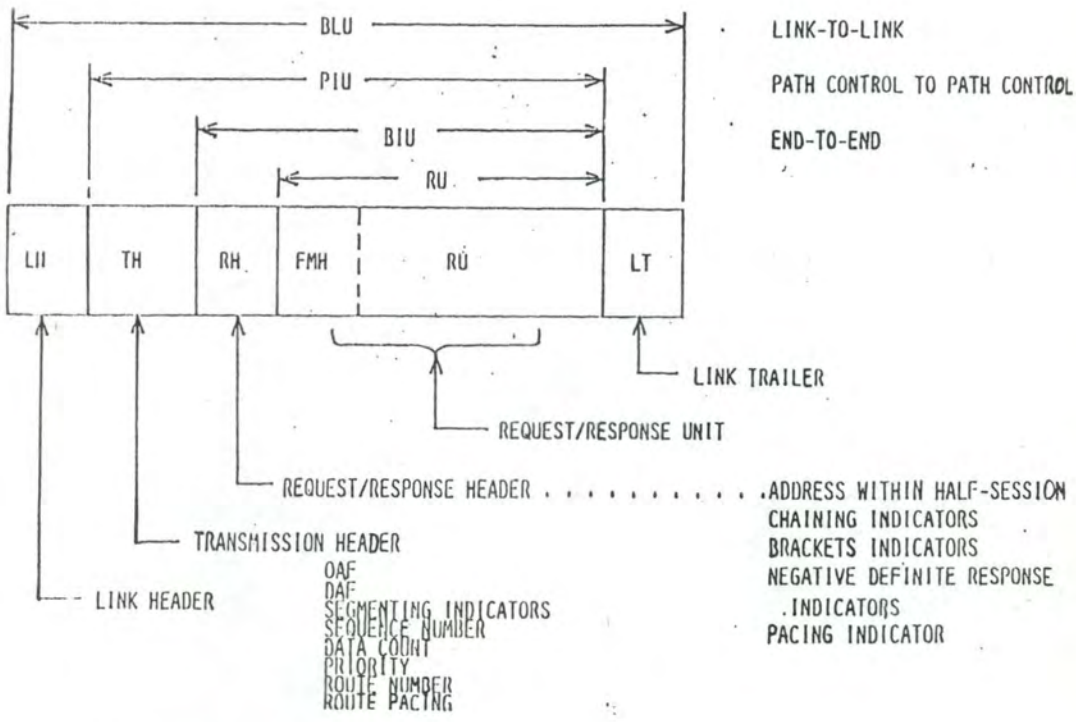


Figure 21 : message format

resynchronizing session data flow.

- Services provided by the Session Control

It helps to control one particular Half-Session (NAU).

Manages the activation/deactivation of one half session with either LU or PU by using session control commands.

Start or terminate data traffic ,as directed by the EU at the primary NAU, once the session is established.

Helps to higher level resynchronization if the session gets in trouble.

Provides common session control, used when no sessions are underways.

3. Network Control :

Provides means for CPM and Path Control to communicate through the Common Network using already established session between NAUs.

- services provided by the Network Control

Aids in controlling the node's PU and is used in notification of link status and reconfiguration.

This service is not yet fully architectured.

4. OSI equivalents :

These services correspond to the following of OSI :

Session Layer :

- Establishment and termination of session-connection.
- Normal data exchange.
- Context management (interaction management).
- Exeption reporting.
- Recovery.
- Session identification.
- Synchronization.

Transport layer :

- End-to-end flow control.
- End-to-end sequence control.

3.3.4.2.2. Path Control

1) Objectives :

Path Control is a sublayer of the Transmission System.

It manages the shared link resources of the Common Network and routes the messages (Path Information Units), through the intermediary nodes, until they reach the Path Control Element in the destination node. It is thus aware of NAU locations. (One PCE exists per node)

2) Functions (fig. 22) :

- Addresss translation.
- Destination routing.
- Global flow control.
- Link management.

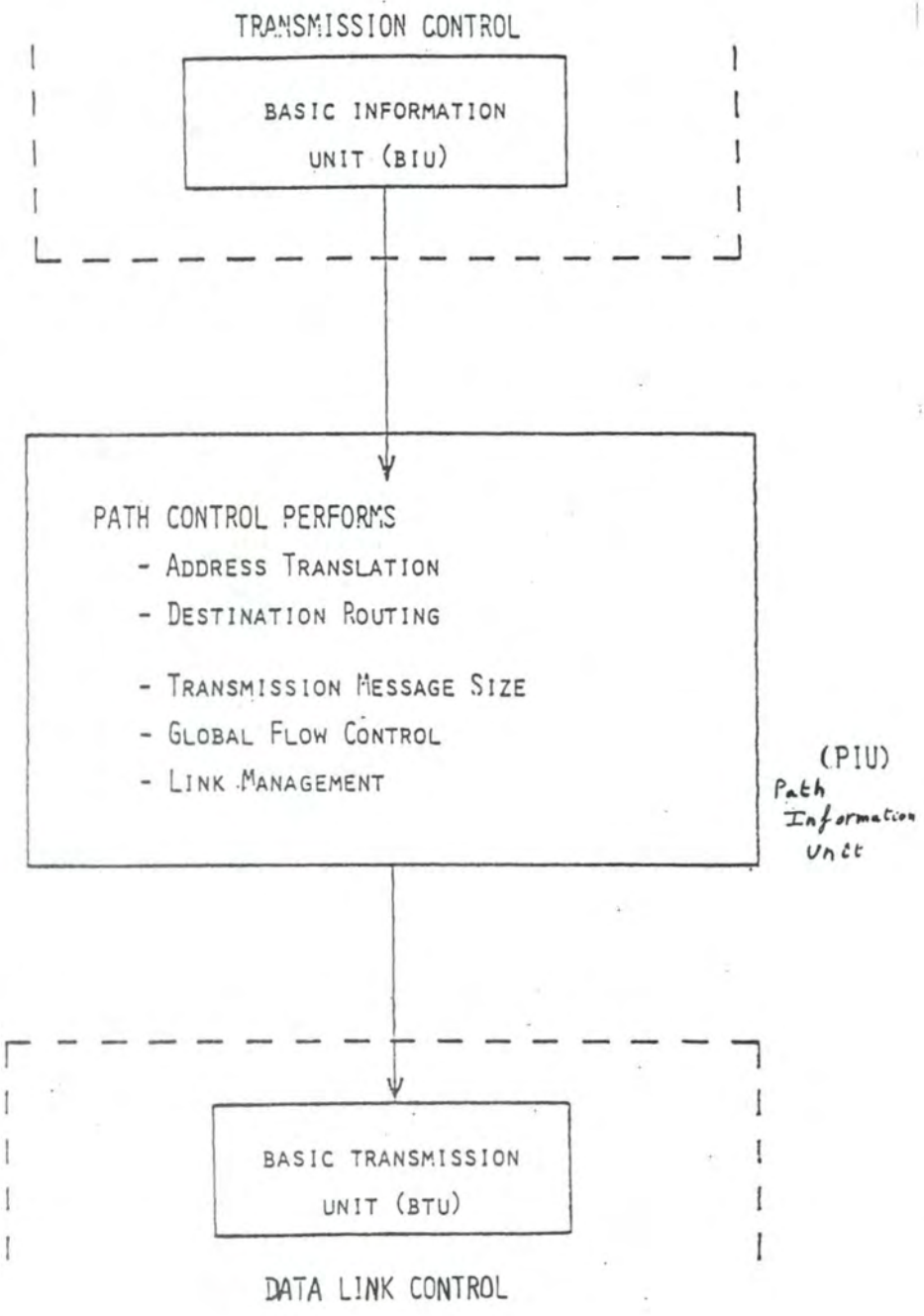


Figure 22: PATH CONTROL FUNCTIONS

- Message size formatting.

3) Services provided :

- Segmentation of BIU received from Transmission Control if necessary, depending on magnitude of the BIU and the accomodation of transmitted unit sizes to buffer size at target node.

- Generation of Transmission Header (TH) containing control information for addressing, mapping, segmenting, sequencing and merging of TH and BIU to make a PIU (Path Information Unit) (fig. 21). These control informations originate from the CPM (Connection Point Manager) in the TC services from PCE.

- Optionally blockage of PIUs in a BTU (Basic Transmission Unit) passed to the Data Link Control. (see fig.x.x for headers and messages formats)

- Masking Primary-Secondary relationships existing between DLC elements and, thus, providing a full duplex path between NAUs.

Those services correspond to the segmenting and blocking ones of OSI Transport Layer.

4) Composition :

The path control in a PU 4 and 5 may be further divided into three sublayers which are the grouping of function s to manage parallel links, transit routing and end-to-end connections.
Those are Virtual Route, Explicite Route and transmission Group controls.

a) Virtual Routes :

A Virtual Route (VR) is a full duplex logical connection between two subarea nodes (CUCNs)/NAUs and only indirectly refers to physical connections (fig. 23).

It is also an ordered list of subareas from

the source subarea to the destination subarea. In fig. 23, ABCF, ADEF, and ADECF are all Virtual routes from subarea A to subarea F. [Tanenbaum 81] A VR corresponds to one or multiple sessions.

It is defined by

1. Subarea addresses of the two ends of the VR.
2. A VR number.
3. A transmission priority.

Services provided :

- Provides dynamic end-to-end flow-control (pacing) and sequencing in cooperation with ER control. Flow control may be different from one VR to the other.
- establishment/termination of a VR between subarea CUCNs for sessions purposes (by activation/deactivation of ERs.).
- Choice of a class of services allowing different Transmission priorities.
- Error detection and recovery (abortion of VR if all ERs fail)
- Upward multiplexing : several VRs achieving different flow controls and transmission priority may use the same ER.

b) Explicite Routes :

An Explicite Route (ER) is a specific sequence of Transmission Groups and intermediary CUCNs connecting two end-nodes hosting the EUs (LUs) wishing to communicate. For exemple in fig. 23, 5683, 5783, 5684, and 5784 are all ERs for the VR ADECF.

It implements the physical representation of

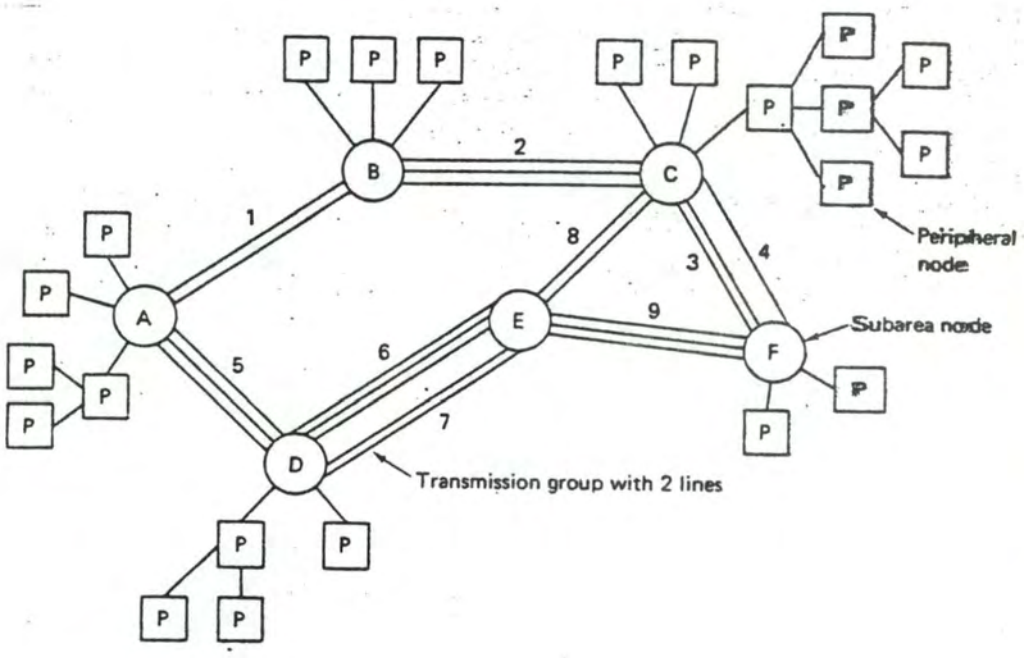


Figure 23 : Explicite Routes

the path between the two end-nodes.

It is defined by

1. Addresses of the two subarea end-nodes of the ER.

2. An ER number.

Services provided :

- Provides alternate routing in case of failure of one ER physical component between two end-components.
  Up to 8 ER are allowable between the end-components.

- ER establishment/termination and notification of failure to the SSCPs and LUs concerned if none ER can be activated or all have failed.

c) Transmission Groups (fig. 24) :

A TG represents one connection or logical link between adjacent communication controller nodes (CUCNs).
This connection constitutes of a set of one or more operating links ruled by their own SDLC protocols.
It provides the function of parallel links similar to the OSI multi-link (1).

It is defined by

1. Addresses of the two adjacent communication nodes.

2. The TG number.

Services provided (2) :

---

(1) : See ISO TC97 SC6 for more information about multi-links.

(2) : For more information see SNA IACF, AHUJA 79 or GRAY-NEIL

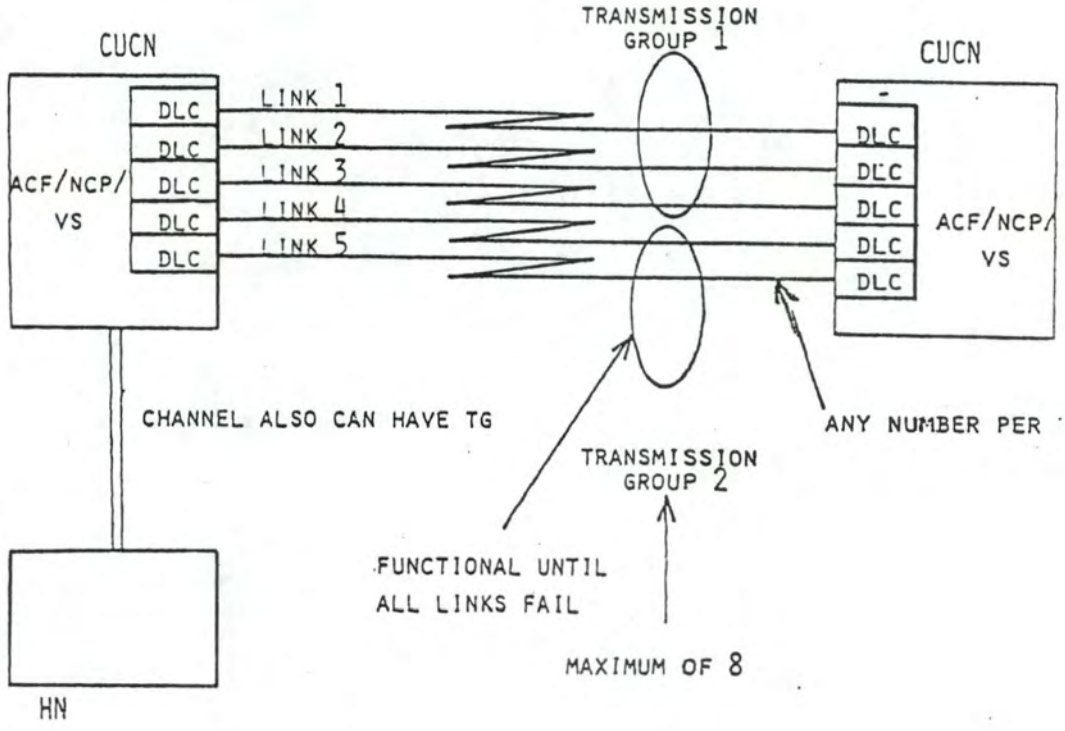


Figure 24: TRANSMISSION GROUPS

- Provides increased bandwidth availability and reliability of logical link between adjacent communication cotroller nodes. It has no end-to-end significance.

- Provides automatic rerouting on an operational link of the TG in case of failure of the ones used.

- It handles the flow of messages over each TG and assures sequential delivery by allowing reordering at the receiving node, error detection, and retransmission of eronous messages. (OSI correspondent could be Network error control and error free transmission.)

  Three levels of message priority are permitted so that higher priority messages bypass the others on each TG.

OSI equivalent :

Globally the Path Control provides a part of the services of the Transport layer and the Network layer :

- Transport layer

  - Establishment/termination of transport connections.

  - Error detection and indication.

  - Upward multiplexing.

  - Mapping Transport address onto Network address.

  - Class of services selection.

  - End-to-end flow control.

  - Segmenting/blocking.

- Network layer :

- Establishment/termination of network connections.
- Routing.
- Data transfer.
- Upward multiplexing onto data link.
- Downward multiplexing.
- Segmenting and blocking functions.

3.3.4.3. Data Link Control

3.3.4.3.1. Objectives

The DLC lies between Path Control (PC) and Physical control. It manages the links attached to the hosting node composed of DLC element (hardware and software) managing one Data Link and functioning as either primary or secondary station depending on the Physical configuration.

The protocol used is SDLC a subset of HDLC, allowing Unbalanced operations on a Normal Response Mode plus some optional functions.

3.3.4.3.2. Services provided by the DLC :

1. Link connection/disconnection between two stations. One acting as a primary, the other as a secondary (the primary leading the communication).
2. Maintaining of synchronization once the connection is established.

   Abortion of the connection in case of unrecoverable errors.
3. Bit stuffing.

4. Error detection and recovery.

5. Sequencing by numbering of frames to be transmitted and control of frames received.

6. Flow control assuring correct handling functions at two end-stations.

3.3.4.3.3. OSI equivalent :

The DLC layer achieves in fact the standards UM/NRM services of OSI Data Link Layer.

3.3.4.4. Physical Control level

Physical Control is not separately defined in SNA but is present beneath the DLC. SNA PC is functionally equivalent to the corresponding level 1 of OSI. It currently implement V24 and X21 standards.

Physical connection is called a link connection and provides a two-way communication between two or more link stations. Links can be permanent or dynamically set up/set down, are duplex or half-duplex, point-to-point or multipoint.

4. DNA DESCRIPTION

4.1. Introduction to DNA

DECnet [Wecker,1980] is a set of programs, protocols and hardware produced by Digital Equipment Corporation. The architecture of DECnet is called DNA (Digital Network Architecture).

The intention of DECnet is to allow any DEC's customers to set up a private network. A DECnet is a collection of machines (called nodes), with their O.S. and software modules, some of which may run users programs, some of which may do packet switching or batch. The functions performed by any given machine may even change in time.

DNA consists of a model, a set of interfaces, and a set of protocols. The DNA model describes a structure that embraces the software modules which perform networking functions for each DEC O.S. . The structure is layered and conforms for a major part to the OSI architecture.

4.2. System cuts

DEC network links computers running different (but compatible) Operating Systems. The figure 25 showns a six nodes meshed network.

The DECnet implementation at each node acts as an interface between the node's O.S. and the network (fig. 26) converting DECnet formats to those recognizable by node's O.S., and reverse.

DNA appears thus to be a nodes' distributed network, topologically hostless. In terms of System cuts those identified are :

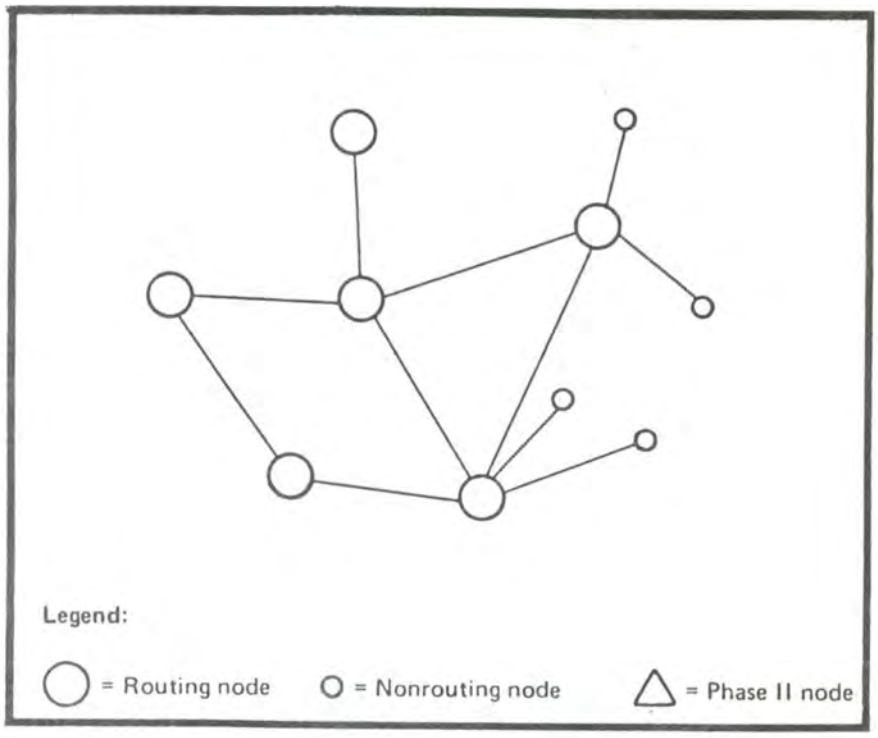


Figure 25 : A 6 Nodes Configuration

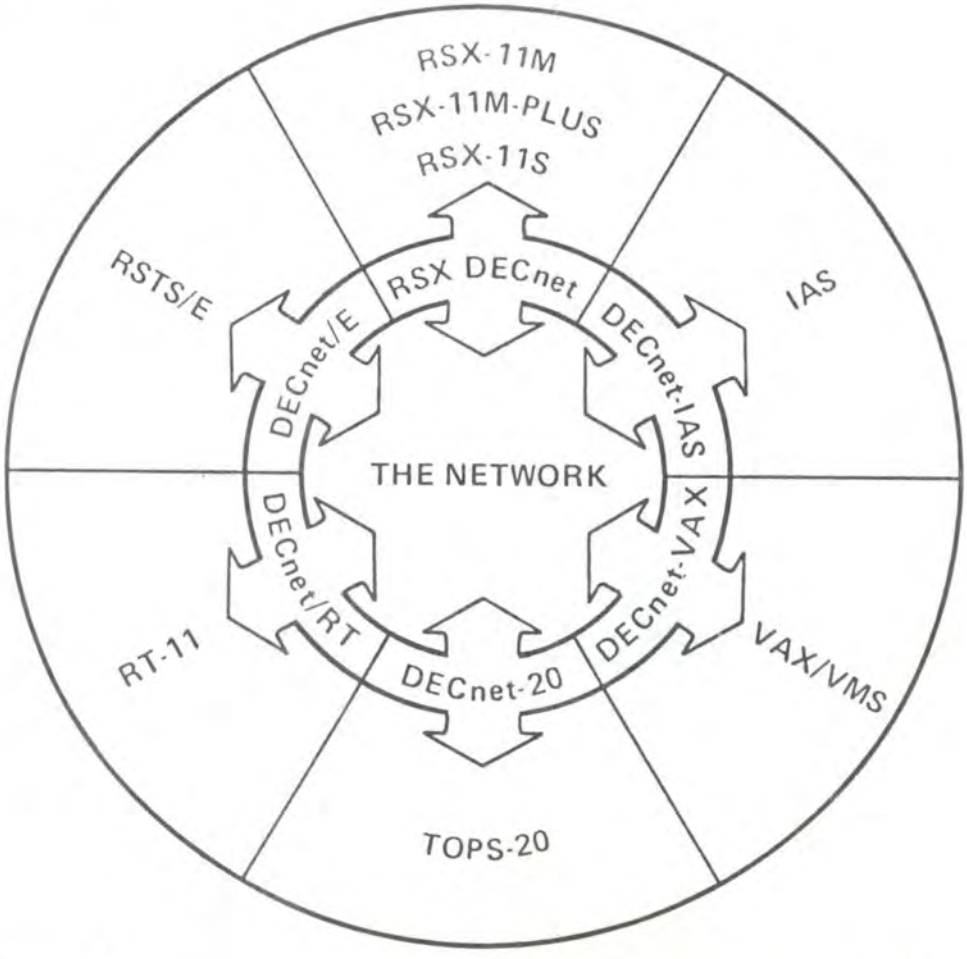


Figure 26 : DecNet Interaction with various O.S.

4.2.1. Nodes :

Nodes are DEC computers running their own O.S. which allow interactions through DECnet.

DEC doesn't identify nodes devoted to data processing and others devoted to transport functions. A node assumes the two functions, acting as a Data processor for local applications (80% of resources) and as network processor assuming networking functions (20% of resources).

In this context three subtypes of nodes are distinguished :

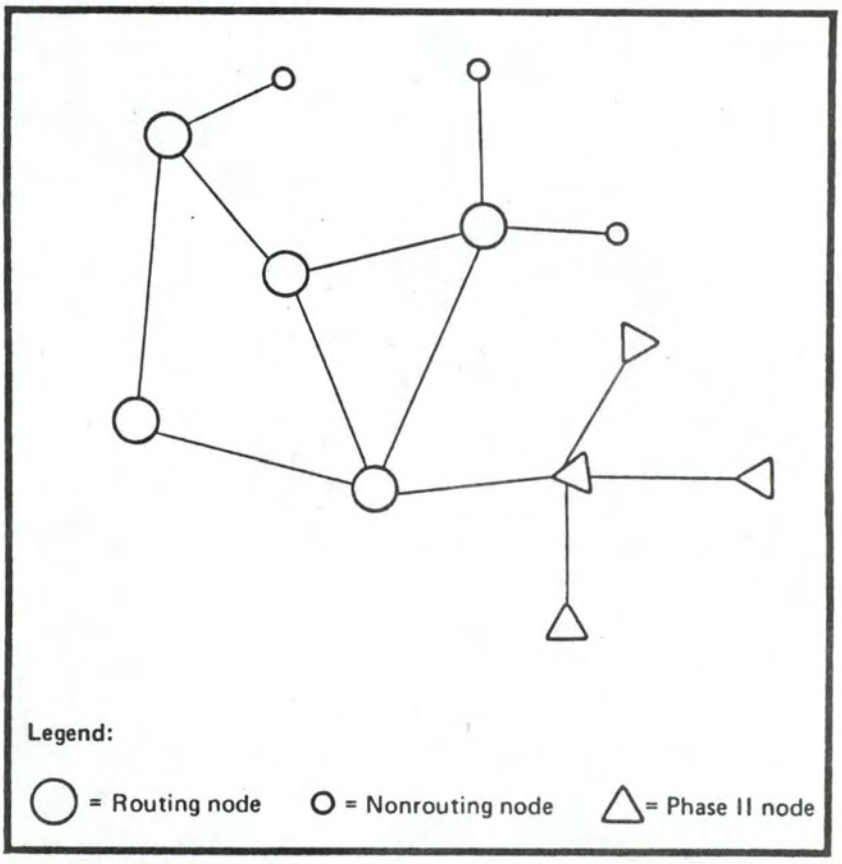
1. Routing nodes : a routing node can forward packets to other nodes in the network and can be adjacent to all other types of nodes.

2. Nonrouting nodes : a nonrouting node can sends packets to other nodes in the network but packets can not be forwarded or routed through it. It can be adjacent to one other node only, and is therefor an end-system in a configuration.

3. Phase-II node : runs a previous phase-II implementation of DECnet and therefore does not support full routing. It can send packets only to adjacent nodes and cannot forward packets it receives onto other non-adjacent nodes it the network. It can be adjacent to one or more full routing nodes and/or to other phase-II nodes. Logically it is an end-system node in a Phase-III configuration (fig. 2 7).

- Remarks on network philosophy :

As above mentioned DNA is totally distributed . That is, there is no inherent central control functions (i.e. SSCP in SNA) in the network. To achieve this topological independence, the control and maintenance functions are executed at the level of usr's applications within the DNA structure (Network



Phase III nodes cannot communicate with the Phase II satellite nodes.

Figure 27 : A Mixed Configuration: a Phase III Network Adjacent to a Phase II Star-shaped Network

Management Layer).

A second characteristic is that all nodes are addressed uniformly. The network has no inherent notions of a backbone communication network (i.e. no PU types as in SNA). The notions of host nodes, concentrator, and communication switching nodes are logical ones and depend on the software, as said before. Two nodes can change from host-host relationship to a host-FEP relationship without affecting the user or network software. The transport level communication protocol and addressing are the same for both these situations.

These characteristics are achieved by having DNA built around the following principle : all network usage can be modeled as communication between application level processes. These application level processes are called resource objects and may be application programs (tasks), operator or I/O devices. To this end, there are several ways in which computers in a network can work together (fig. 28).

- In the program-to-program mode , a program in one node or computer requests a program in another to perform a data processing task, and the result are retained. That often involves gaining access to a data file.

- In the file-transfer mode, the first computer retrieves a data files from a storage device located at the second computer and computes locally. The same computing task can be performed in either mode.

- The third mode in which computers can work together, resource access, describes their ability to share network resources files, line printers, terminals, graphic plotters, and application programs. Typically, resources access permits one computer to retrieve a single record from a disk file linked to another computer as if the file belonged to the first computer. The first computer may requests that a line printer, controlled by the second computer, print out a report. If instead, the first computer wants the second to create a report file and return that file in the file transfer mode to the local site for printing, it can request this in the program-to-program mode..

No matter what type of traffic flow (interactive, real-time, or batch is occuring, the flow is always in one of these three modes.

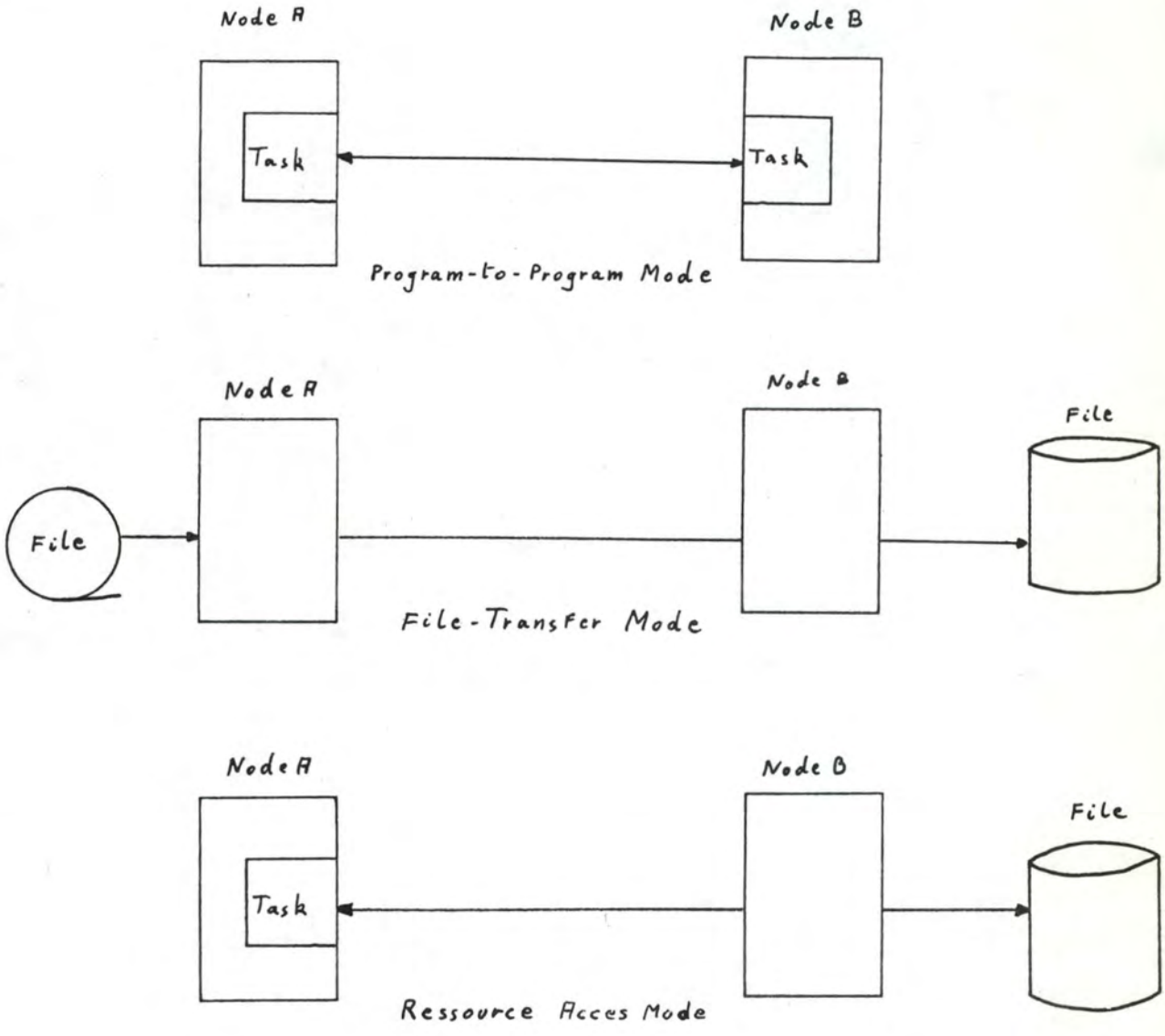


Figure 28. Resource Object Modes

- Configuration :

A DNA network [DNA DTP] consist of two or more DEC processor nodes each loaded with DECnet software product compatible with its operating system. Each DNA interprocessor operation utilizes a layered architecture and a common set of DECnet protocols. The range of functions that can be performed between any two nodes is limited to the functions they share (the network as a whole, however, is not limited to the functions common to all).

4.3. Service Cuts

4.3.1. DNA layering structure

The layering decomposition of DNA is that shown in figure 29. It consists of six major functional layers plus a Network Management Layer.

4.3.2. Application Layer

It encompasses user writen programs and services that access the network.

Underlying layers provide it with a set of networking functions :

- Task-to-task communication (program-to-program): for the exchange of data, remote processing requests, dialogue between two application progams.
- Remote file access and transfer : to access, work on, request the transfer of files or records of files from remote locations.
- Resources access : for network resources sharing (files, line printers, terminals, programs, ...) as if they were local resouces of the application layer.

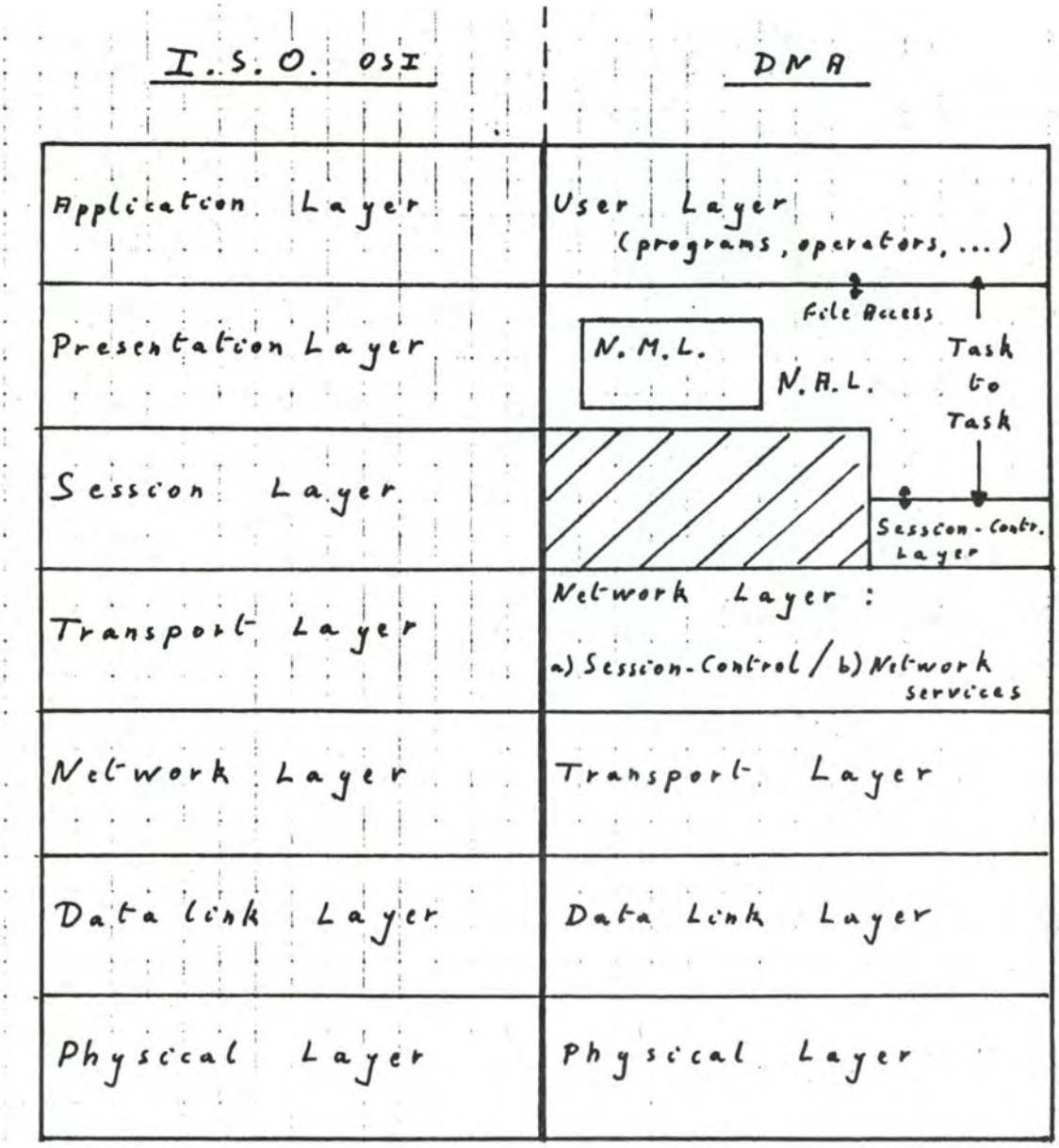
4.3.3. Network Management Layer

4.3.3.1. Objectives

Special purpose layer. It allows system managers (using NCP (1)) to control and monitor network operations, and provides informations for network

---

(1) : " the Network Control Program (NCP) is a DECnet utili-



N.M.L. : Network Management Layer

N.A.L. : Network Application Layer

Figure 29 : DNA Layering Decomposition

evolution planning and problems correction.

As it is defined in [DEC NMFS]: " it is the only layer that has direct access to each lower layer for control purposes (fig. 30). Modules in this layer provide user control over, and access to, network parameters and counters. It also performs up-line dumping (remote memory dumping), down-line loading (remote memory loading), and testing functions."

It is not a layer in itself but more a set of primitives functions or tools that are present in each other layers for networking management (the kernel being the N.M.L.). Even so, Network Management can be distributed along the nodes to achieve a part of the network control ("control over a DECnet network can be either distributed or central. Distribution of control can be either partial or complete [DEC GD]").

The services provided are performed using the Network Information and Control Exchange (NICE) protocol and Maintenance Operation Protocol (MOP) for communication purposes.

4.3.3.2. Functions

1. loading and dumping remote systems : to load an O.S. into a remote node or configure it for network purposes.

2. changing and examining network parameters : for exemple, an operator can change line costs or nodes names.

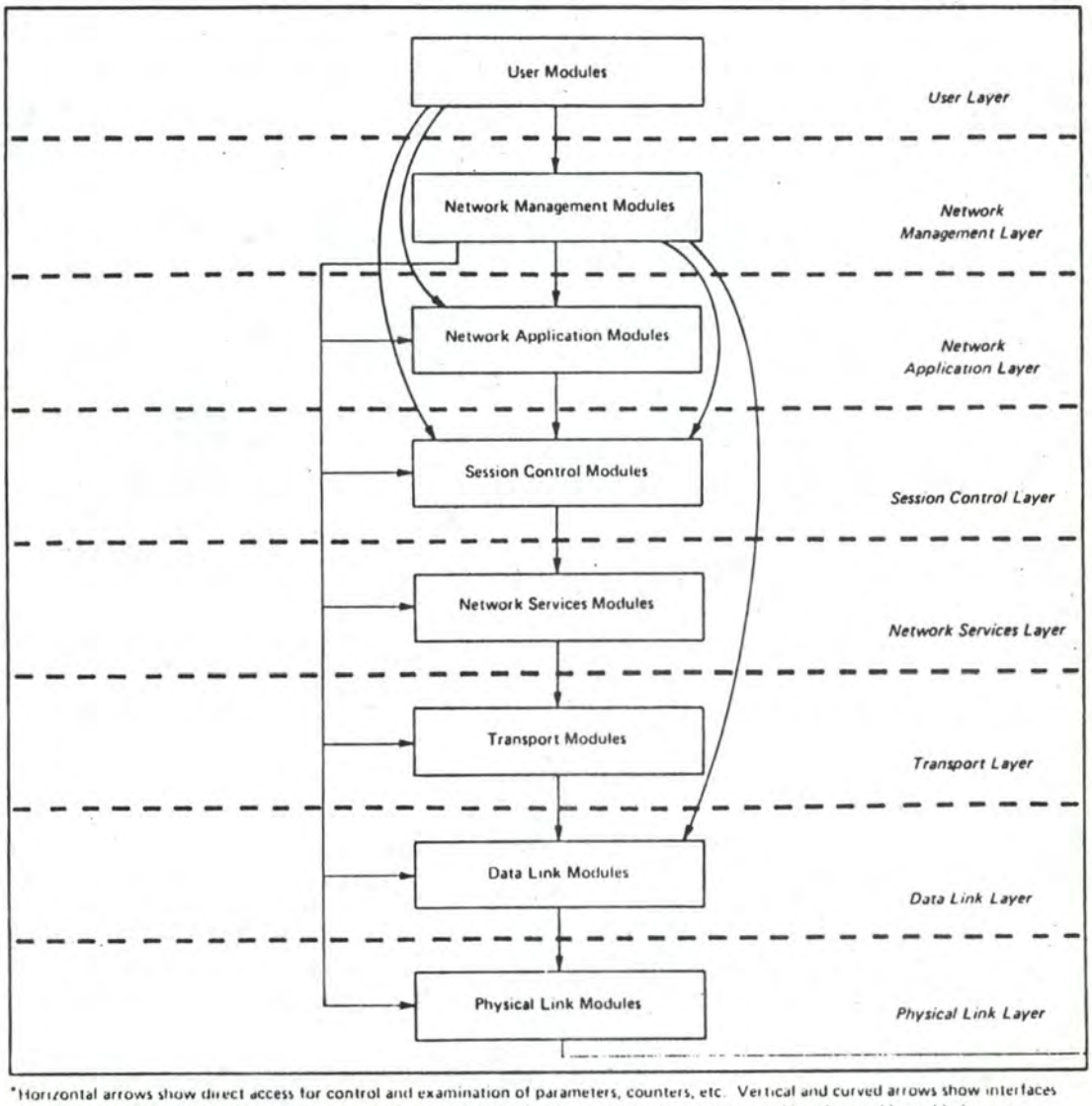
3. examining network counters and events that indicate how the network is performing.

4. testing links at both data link and logical link levels.

5. setting and displaying the states of lines nodes : for reconfiguration by turning lines and nodes on or off.

---

ty program that accept terminal commands to load, monitor and test DECnet software."



*Horizontal arrows show direct access for control and examination of parameters, counters, etc. Vertical and curved arrows show interfaces between layers for normal user operations such as file access, down line load, up line dump, end to end looping, and logical link usage

Figure 30 Relationship of Network Management to Other DNA Layers

4.3.3.3. Composition (1): (fig. 31)

Briefly we can say that N.M. components are as follows :

- at user layer : the NCP interfaces with other layers and provides a standard set of commands.
- at N.M.L. : the main routines and modules for management functions, reception of commands from other nodes and their execution, taking into account events occuring in other layers, manager services links.
- at Network Application level : link test routines.

4.3.3.4. Services provided to users

- bringing up and down a system/ a node/ a network .
- monitoring of local or remote nodes.
- testing network components (i.e. data and logical links, ports, ...).
- modifying a network configuration.

4.3.3.5. ISO equivalent

No OSI correspondent except the System-management-application-processes in the Application Layer.

---

(1): for more informations look at DEC GD, DEC NMFS, and DEC SMG.

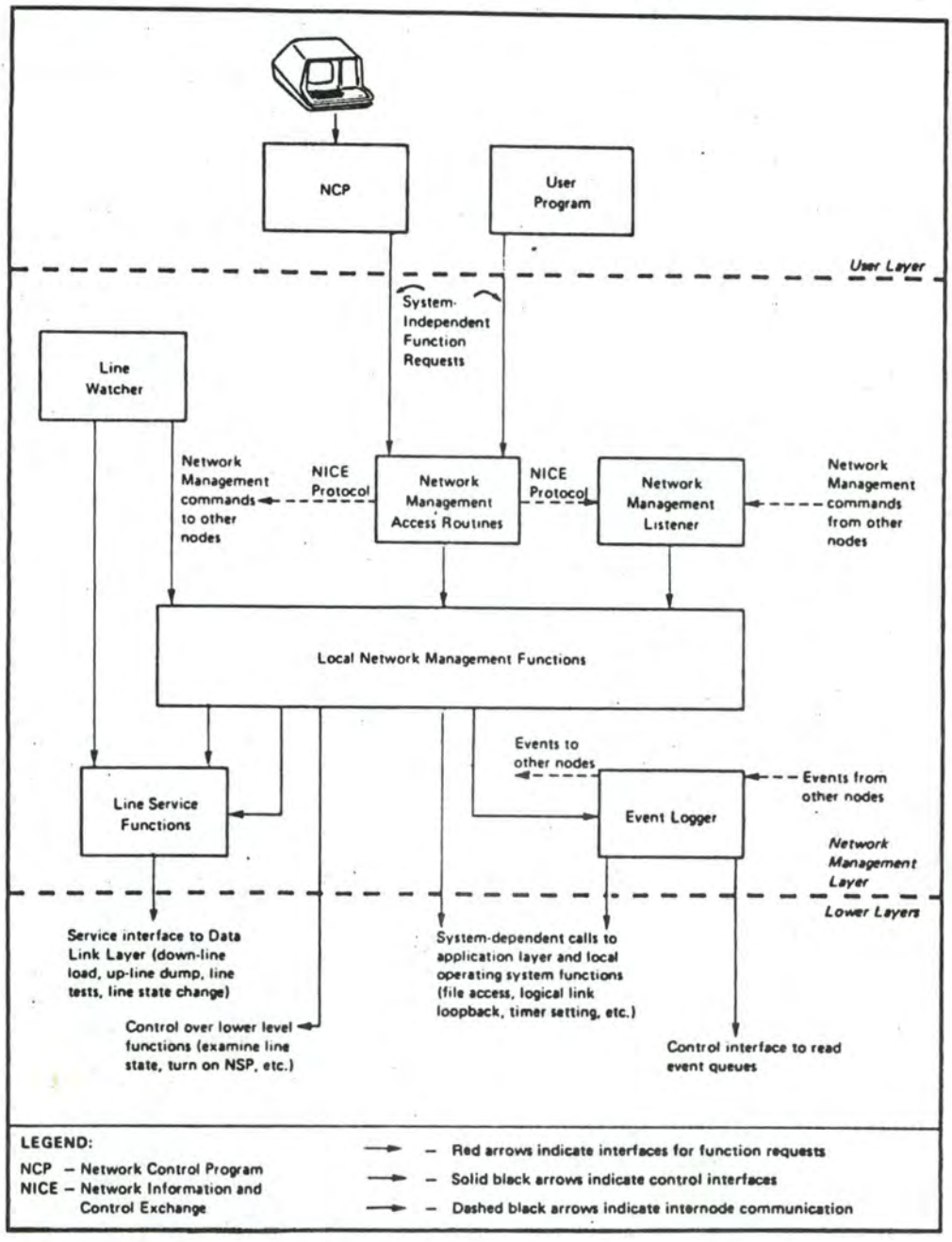


Figure 31 .: Interrelationship of Network Management Components at a Single Node

4.3.4. Network Application Layer (N.A.L.)

4.3.4.1. Objectives

N.A.L. is responsible for remote file access, remote file transfer, and resources managing programs. This layer contains both end-user and DEC-supplied modules which execute simultaneously and independently.

4.3.4.2. Composition

It includes two protocol services :

1. Data Access Protocol (DAP) : for remote file access and handling.
2. Loopback mirror protocol : (not described here) for logical links testing by the N.M.L.

4.3.4.3. Services provided

DAP is not only a file transfer protocol but something more general allowing file manipulation.

DAP provides user with

- remote file access and manipulation (opening/ deleting/ access to records/ modifications/ ...)
- remote file transfer utility insuring error free communication and reports of fatal error, and handling formats conversions.
- network command terminal function (analog to virtual terminal services): local user can log onto a node in the network as it is directly connected to it.

4.3.4.4. Functions

- supports heterogenous file systems.
- permits sequential, random, and indexed record access.
- allows command files to be handled.
- all operations on files.
- upward multiplexing of logical links.

4.3.4.5. ISO equivalent

This layer includes a part of the OSI presentation layer, but not advanced features like encryption or compression of text. It provides only some file format conversions and the above mentioned services. It may be viewed as an application oriented presentation layer and not as a multipurposes P.L.

4.3.5. Session Control Layer and Network Control Layer

4.3.5.1. Objectives

As defined by DEC :

- "The Session Control (S.C.) defines the system-dependent aspects of Logical Link (1) communication. S.C. provides functions such as name-to-address translation, process addressing, ..."
- "The Network Services (N.S.) defines the system-independent aspects of Logical Link

---

(1): a logical link also called virtual link by DEC, includes the notions of both session-connection and transport-connection defined by ISO. It is an individual temporary end-to-end connection between two upper layer entities (here, modules in N.A.L. or application layer) allowing them to exchange data for the duration of the connection.

communication. They enable the creation, maintenance, and destruction of logical links, data flow control and end-to-end error control, segmentation and reassembling of messages."

These two layers work together to allow users (2) (programs or operators) to communicate, through the network, via an individual Logical Link, regardless of the network users locations.

The purpose of Session Control is to bridge the gap between end-users requiring logical link service, unaware of locations problems, and the network services which actually create, maintain, and destroy these links (fig. 32).

4.3.5.2. Functions and composition

a) Session Control (3)

1. functions :

- mapping node names to node address (and reverse) using mapping tables.
- identifying end-users (4): determines if an existing end-user corresponds to the destination end-user specified in incoming connect requests.
- activating or creating processes for command handling.
- validating incoming requests.
- processing connect request from end-users :

---

(2): users are upper layer entities in the ISO vocabulary.

(3) : for more information look at DEC GD or DEC SCFS .

(4) : may be though of as the session-endpoint-identification in OSI.

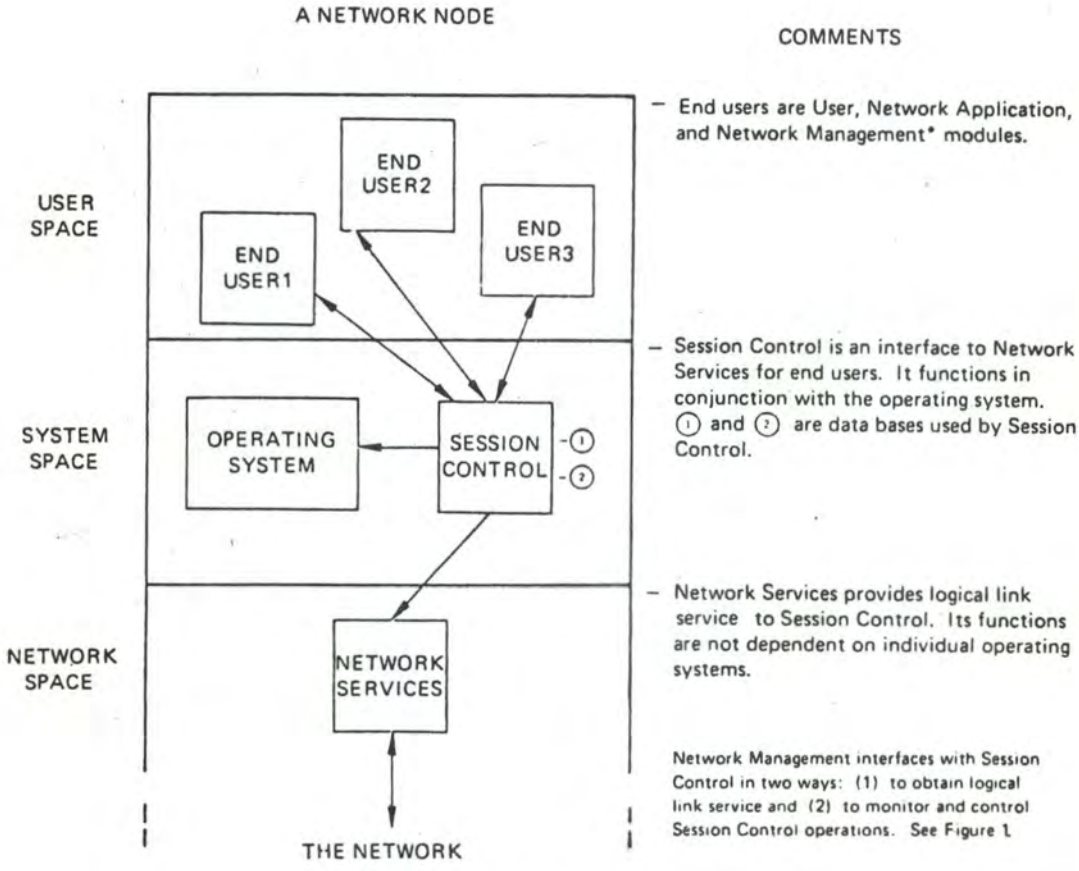


Figure 32 : A Session Control Model

› mapping from destination node name to destination node address

› formating the connect data for Network Services

› issuing a connect request to Network Services

› starting outgoing connection timer

- receiving and processing incoming connect requests from Network Services which implies :

  › to parse connect data to obtain control informations,

  › to validate control information,

  › to identify destination end-user and activate it,

  › to deliver the connect request with source node' name,

  › to start an incoming timer.

- the other functions are directly passed to the N.S.L. .
  They include sending and receiving data, disconnecting and aborting a logical link.

2. <u>interfaces</u> :

The Session Control maintains four interfaces between itself and its environment:

- one to the network services to use the Logical Link service.

- one to the end-users to handle their commands and passe their data.

- one to Network Management for mapping tables, session control and modifications.

- one to the O.S. for end-users monitoring.

3. remark : Session Control represents the point at which DECnet is integrated within an O.S.. It can not be specified in isolation from non DECnet modules. Indeed, the end-users are created and managed by the O.S.. For this reason the O.S. interface has to exist.

b) Network Services

They provide a process-to-process (1) communication servicethat allows two processe to exchange data reliably and sequentially, regardless of their locations in a network. A connection between two processes is called a Logical Link.

1. Logical Link Services

- creation, maintenance, disconnection and abortion of logical links.

- delivery of data and control messages, in sequence, at the proper destination via two subchannels constituting the logical link :

  a) a normal data flow service to convey normal data messages.

  b) an expedited data flow service for acknowledgment and control data (such as interrupt messages, data request messages, ..).

- segmentation and reassembly of data: messages are segmented and numbered if they overcome the transport layer data accepted size.
  At reception the segments are reassembled in correct sequence.
  This service applies only to normal data.

- error control : by a mechanism of acknowledgment, time-out, and retransmission of bad data (fig. 33).

---

(1) : the notion of process here is similar to the one of session-entity or application-entity of OSI.

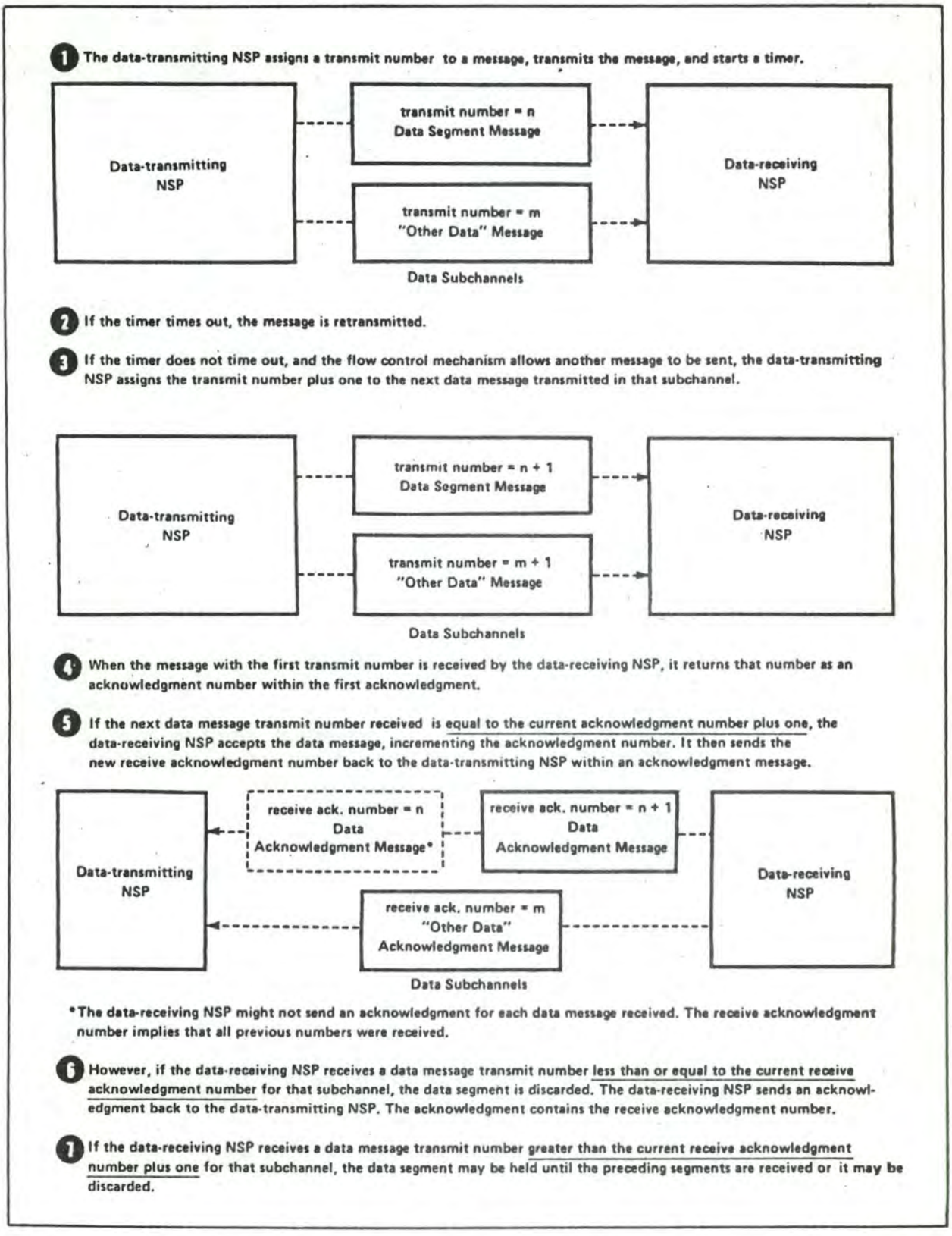


Figure 33 : Acknowledgment Operation

- flow control : to ensure that data is not lost for lack of buffering capability, and that deadlocks do not occur. Both normal and expedited data are flow-controled, but on independent basis.

  The normal flow control may be selectionned amid three flow control types :

  1) none : flow is managed by upper users.

  2) segment : receiver regulates the number of segments it accepts to receive.

  3) messages : here the number of messages to receive is regulated.

2. Interfaces :

The N.S. maintains three interfaces to its environment:

- session control interface to provide Session Control with logical link service.
- network management interface for the control of the N.S. by the N.M.L.
- transport interface to use the services of the transport layer (network OSI layer) (i.e. sending and receiving datagrams to/from any N.S.L. in the network.

4.3.5.3. Services provided to upper layer

a) Session Control Layer

Provides end-users with process-to-process communication functions:

› establishment/ disconnection/ abortion of connections between end-users.

› error free data transfer.

› end-user name resolution.

b) Network Service Layer

Provides the S.C.L. with logical link service .

4.3.5.4. ISO equivalent

These layers globaly provide the same services and assume the same function that the OSI Transport Layer and a part of the Session Layer.

1. Transport layer

› end-to-end flow control

› segmentation

› error control

› upward connection multiplexing

› addressing

› transport connection identification

› sequencing

2. Session Layer

› dialogue management (TWS)

› mapping session connection onto transport connection

› establishment/ end of session connection

› normal and expedited data transfer

4.3.6. Transport Layer (ISO network level)

4.3.6.1. Objectives [DEC GD]

It is a message delivery service.

Transport accepts messages, called packets (1) in the context of transport, from the N.S.L. in a source node's transport entity and forwards the packet, possibly through intermediate nodes to a destination node's transport entity.

Transport implements a datagram service which delivers packets on a best effort basis. It selects routes based on network topology (finding an alternate path if a path component fails). This service doesn't guarantee delivery of packets to N.S. at the destination node in the same sequence in which they were received from the source N.S.. Transport may duplicate, modify or nisdeliver a packet.

4.3.6.2. Functions [DEC GD and DEC TFS]

1. determines packet paths. A path is the sequence of connected nodes between a source node and a destination node. If more than one path exists, the best one is determined by a routing function. (fig. 34)

2. forwards packets : to the N.S.L. or to the next line in the path depending on the destination of the packet.

3. manages the characteristics of packet path : transport finds alternate paths, if one exists, in case of a failure of an active one.

4. update process : to make other adjacent nodes aware of routing changes (line down or node up ..)

5. returns packets addressed to unreachable nodes if requested to do so.

6. buffers management in routing nodes for communication purposes.

7. packet lifetime control : to prevent old packets

---

(1) : a packet is a unit of data and control information to be routed from a source node to a destination node. It is a network protocol data-unit in the ISO vocabulary.

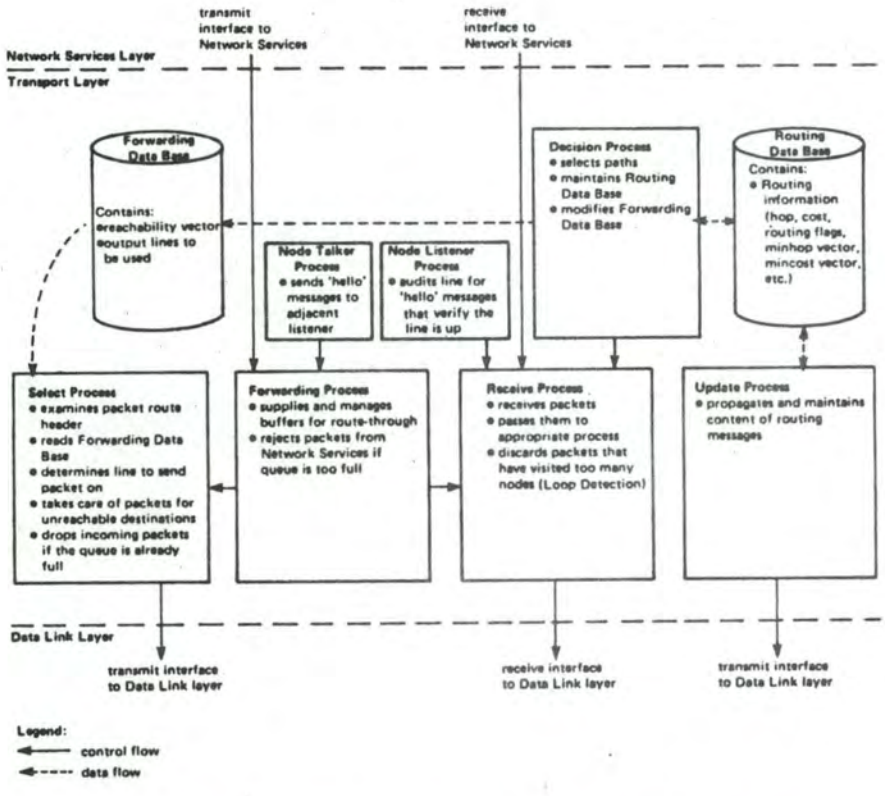


Figure 34 :Transport Components and Their Functions

from cluttering the network.

8. monitors errors reported by Data Link Layer (D.L.L.).

9. keeps track of events for Network Management purposes.

10. delivers packets between phase-II and routing nodes .

4.3.6.3. Composition

The transport layer consists of two sublayers :

a) Transport Control

Supplies full-duplex packet transmission between any pair of nodes. It masks the physical and topological characteristics of the network from high layers. It is independent of the D.L.L. below it.

It consists of the following components

- routing : performs the functions 1, 2, 3, 5, 10.

- congestion control : manages the buffers in each routing node by limiting the maximum number of packets on a queue for a line.

- packet lifetime control : bounding the number of nodes a packet can visit.

b) Transport initialization

Masks the characteristics of the D.L.L. from the T.C. sublayer.

It consists of the following components:

- initialization : to identify adjacent nodes and their Transport Layer and perform node verification.

- Physical line monitor : handles error reported by the D.L.L. .

c) interfaces :

The transport layer maintains three interfaces to its environment :

- Network Management interface : to allow N.M. to control and observe the T.L. .
- data link layer interface : to handle commands to and responses from the D.L.L. (data or error reporting).
- N.S.L. interface : to handle commands from and responses to the N.S.L. .

d) note about routing :

1. Routing exemples are illustred in Appendix D of [DEC TFS] p 54.
2. Routing description is developped in Protocol cuts section.
3. For more details consult
   [Tanenbaum 81] p 235 - 237
   [Datapro] C11 - 384 - 108 ; row 2
   [DEC TFS] p 9, 23 - 33, Appendix A, B, C, D

4.3.6.4. Services provided to upper layer

Essentially a datagram service assuming transmission of self-contenent packets through the network, with error report. Packets may or may not be delivered in sequence, or delivered at all; they may be loop, be duplicated, be dicarded by the congestion control.

4.3.6.5. ISO equivalent

1. Transport Layer

   › addressing

2. Network Layer

> routing and switching

> network connection

> upward multiplexing

> flow control

> error notification

> quality of service parameters

4.3.6.6. note

Digital, following an open system politics, will provide X25 and Ethernet interfaces at transport level.

4.3.7. Data Link Control Layer (D.L.L.)

4.3.7.1. Objectives

D.L.L. provides error free communication, and manages lines between adjacent nodes. It is independent from device characteristics.

4.3.7.2. Composition

There are two protocols concerned with the D.L.L.:

1. DDCMP :

Digital Data Communication Management Protocol (DDCMP) is a byte oriented protocol designed to operate over synchronous duplex/ half-duplex channels or data links, switched or dedicated, point-to-point or multipoint data link (fig. 35).

2. Maintenance Operation Protocol (MOP)

Which aims to manage lines resources and maintain communication between adjacent nodes.

4.3.7.3. Functions

The D.L.L. provides the following functions :

- Controls the operation of the physical link between nodes, while maintaining the integrity and sequentiality of transmitted data.
- Sliding window flow control with up to 255 frames outstanding. the maximum frame size allowed is 16383 bytes.
- Operates independently of channel bit width (serial or parallel) and transmission characteristics (asynchronous or synchronous).
- Error detection.
- Retransmission of erronous frames.
- Operates as above mentionned in both half-duplex and full-duplex mode, and supports point-to-point or multipoints configurations.

  Remark :

  When half-duplex lines are being used, one station is implicitely the master until it signals a 'change of turn', at which time the other side can begin to send.
  When multidrop lines are being used, one station is master (control station) the others being slaves (tributaries). All communications is from or to the master; no direct slave-slave communications are allowed. (fig. 35).
- Transparency.
- Sequencing by numbering frames.
- Synchronizes transmission on byte and message level.
- Error repporting.

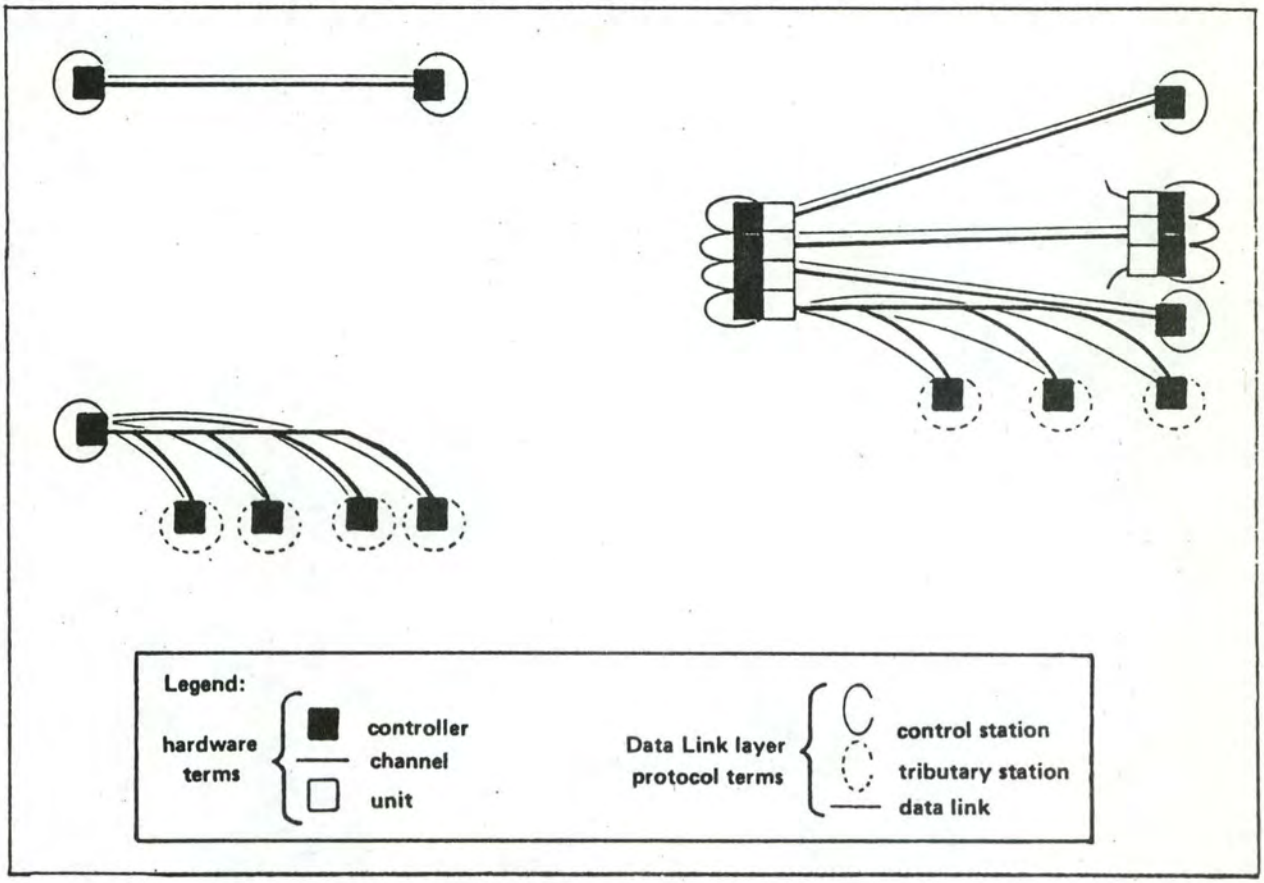


Figure 35 :Link Terminology

- Link management for transmission and reception control, and data flow direction control

- Maintenance mode for tests and bootstrapping functions which use the MOP to perform the following functions :

  - downline loading the memory of a remote adjacent satellite node (1).

  - up-line dumping of memory contents (upon a failure for instance).

  - testing of Data Link and its components.

  - restarting a remote satellite node.

4.3.7.4. Services provided to upper layer

- link connection/ disconnection between two adjacent nodes (stations) .

- synchronization

- abortion of connection in case of unrecoverable errors

- error detection recovery

- flow control and sequencing

4.3.7.5. ISO equivalent

Data link layer :

- data link connection/ disconnection

- data unit transfer

---

(1) : in a configuration, the node being serviced (loaded, The node providing the service is called host node.

- sequencing
- error detection / recovery/ notification
- flow control
- part of service parameters selection
- delimiting and synchronization

4.3.8. Physical Layer

4.3.8.1. Objectives

To realize physical communication between adjacent nodes.

4.3.8.2. Functions

Manages the physical transmission of information over data channel between adjacent nodes.

4.4. Protocol cuts

4.4.1. Introduction

DNA protocols define the relationship between correspondent layers or modules in separates nodes (fig. 36).

A module in one node, communicates with its correspondent in another node via the underlaying layers. Correspondent meaning resident in the same layer and serving the same network functions.

DNA does not defines protocols for all functional layer (fig. 37); at user level for instance protocols are user defined and do not part of the architecture.

DNA allows more than one protocol to exist at one level. For exemple the N.A.L. can include modules using DAP or user's protocols to achieve specific network applications. The user can also substitute his own protocol if equivalent protocols exist in the network.

In the following sections we will describe some of the more important protocols defined in DNA. AN overview of those latters is shown in figure 38.

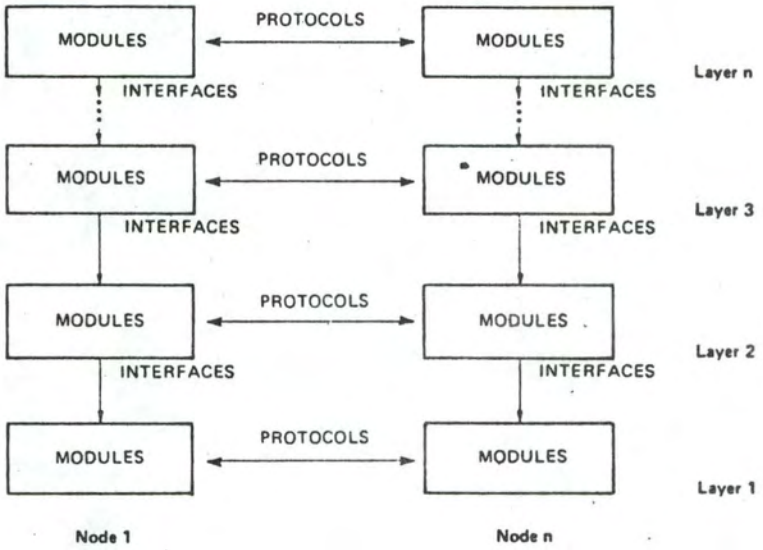


Figure 36 : Basic DNA Structure

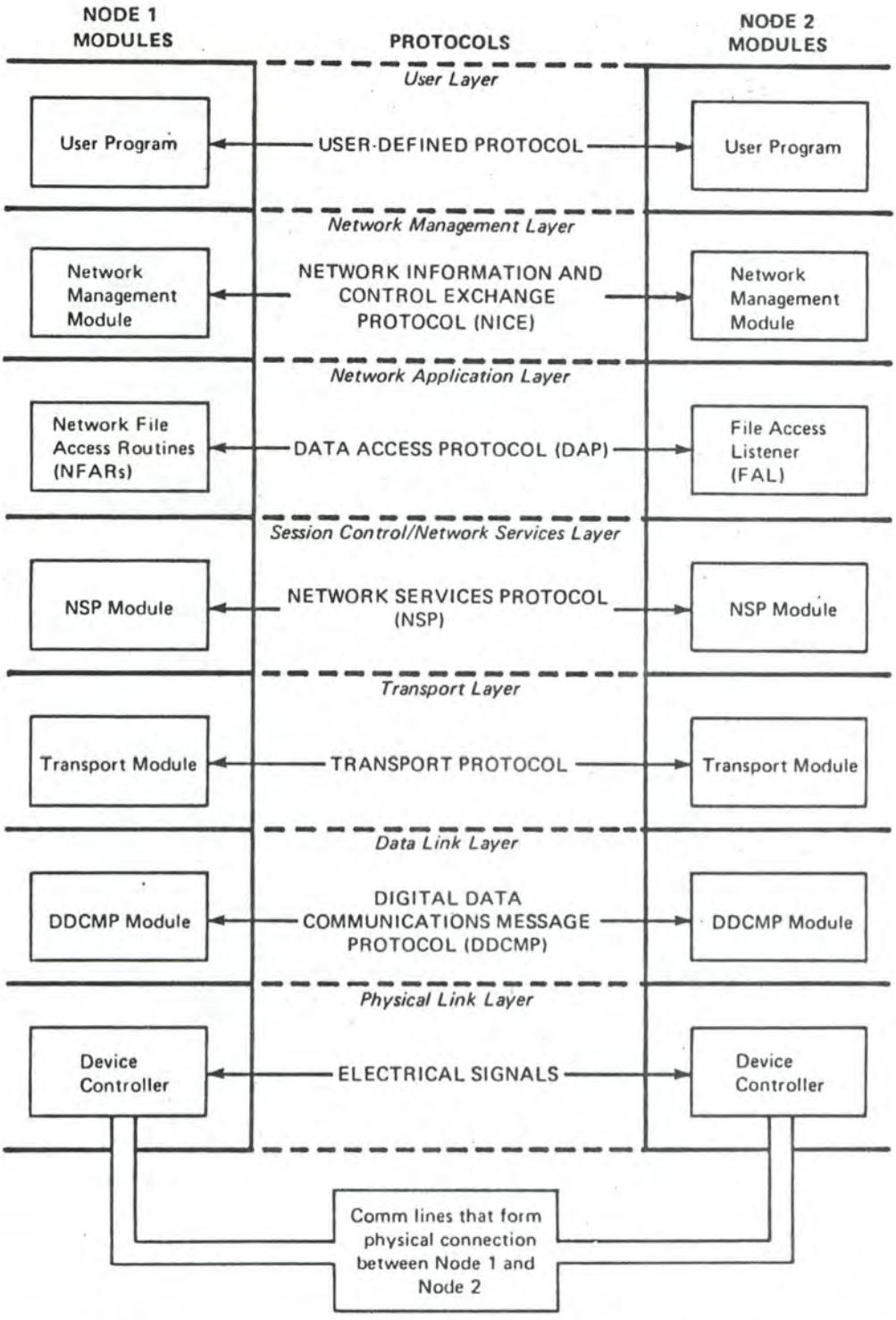


Figure 37: Protocol Communication between Equivalent Modules

Protocol	Layer	Description
NICE	Network Management	The Network Information and Control Exchange protocol defines mechanisms for exchanging network, node, and configuration data, and for servicing requests from modules residing in the Network Management Layer.
DAP	Network Application	The Data Access Protocol defines mechanisms for performing remote file access and remote file transfer on behalf of software modules residing in the Network Management Layer (Phase III only) and the User Layer. See Chapter 6.
NSP	Network Services	The Network Services Protocol defines a mechanism for creating and maintaining logical links between higher-level modules residing in the same node or in different nodes.
Routing	Transport	The routing protocol defines a mechanism for dispatching data to any node in the network by the best possible route. This protocol is implemented in Phase III products only. See section 2.5 and Chapter 3.
MOP	Data Link	The Maintenance Operation Protocol defines mechanisms for transmitting data over a communications channel to achieve specific functions: down-line loading of a remote node; up-line dumping from a remote node; testing a node and network connections; and starting up an unattended remote node.
DDCMP	Data Link	The Digital Data Communications Message Protocol defines a mechanism for ensuring the integrity and sequentiality of data transmitted over a communications channel.

Figure 38. DNA Protocols

4.4.2. Application layer

At this level, protocols are user defined, if exist, and are beyond the scope of this paper.

4.4.3. Network Management Layer

4.4.3.1. Composition

The corresponding entities at this level are composed of

- Network Management Access Routines which provide the Network Management functions.
- Network Management Listener which receives Network Management commands from remote network management entities (fig. 39).

The Network Management entities exchange informations, and control data using the Network Information and Control Exchange Protocol (NICE) (1).

4.4.3.2. NICE protocol

Because coping with Network Manangement tasks, this protocol will be brievly presented.

NICE [DEC NMFS] is a command-response protocol . It includes a set of messages and rules governing their exchange between two peer entities. NICE does not handle error recovery for D.L.L provides error free data delivery.
The messages conveyed by NICE are described in figure 40.

This protocol performs validity checks upon the messages it has to process and forward

---

(1) : two other protocols are used at this level for maintenance and tests (the Event Logger protocol and the Loopback Mirror protocol) but are not laided for they serve too specifics goals. For more information, consult DEC NMFS or DEC GD ch 7.

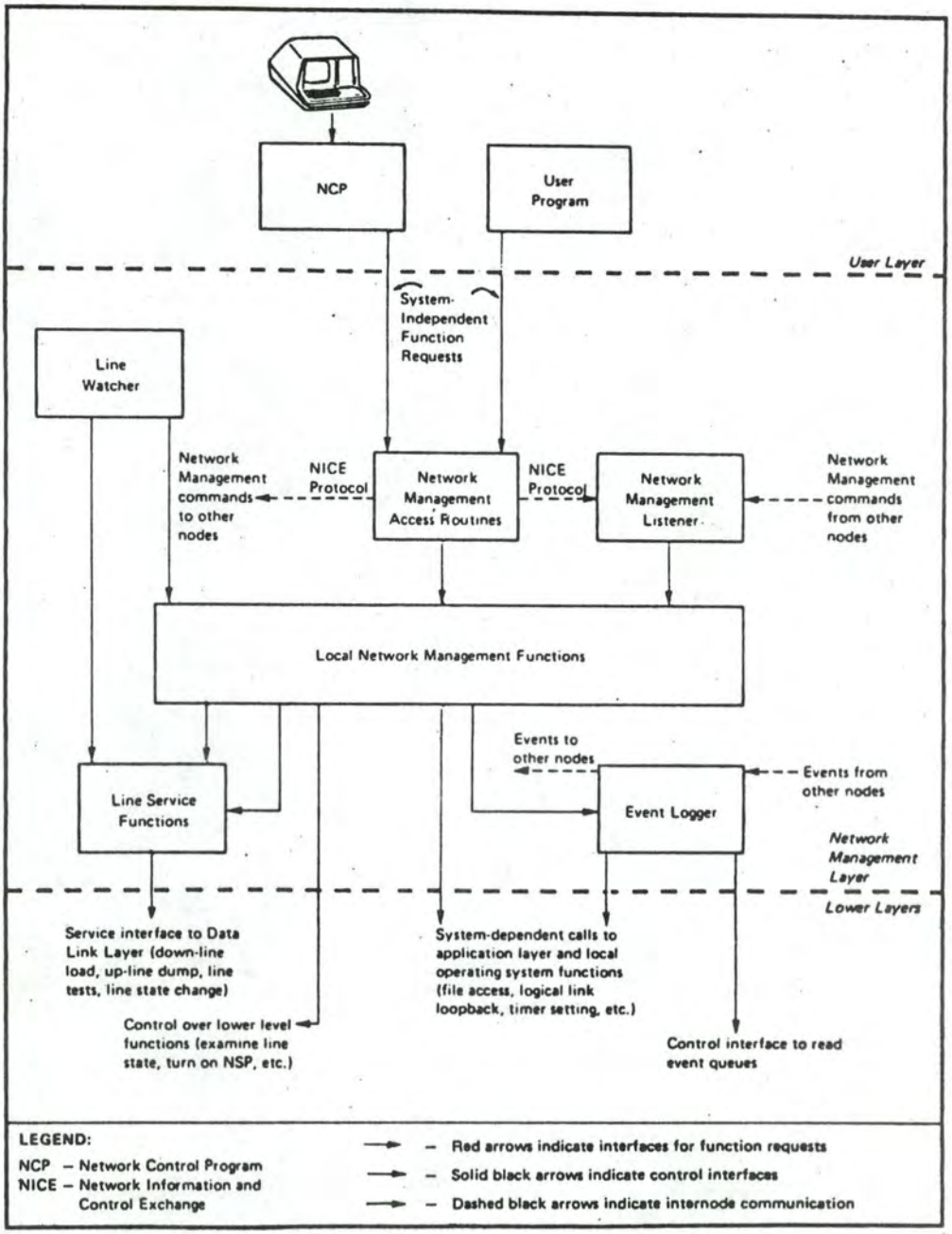


Figure 39 : Interrelationship of Network Management Components at a Single Node

Figure 40 : **NICE Messages**

Message	Description
Request Down-line Load	Requests a specified executor node to down-line load a target node.
Request Up-line Dump	Requests a specified executor node to dump the memory of a target node.
Trigger Bootstrap	Requests a specified executor node to trigger the bootstrap loader of a target node.
Test	Requests a specified executor node to perform a node or line loopback test.
Change Parameter	Requests a specified executor node to set or clear one or more Network Management parameters.
Read Information	Requests a specified executor node to read a specified group of parameters, counters, or events.
Zero Counters	Requests a specified executor node to either read and zero or zero a specified group of line or node counters.
System Specific	Requests a system-specific Network Management function.
Response	Provides request status and requested information in response to a NICE request.

transparently, but has nothing to do with their meaning which is handled by one of both Network Management listener or Network Management Access Routines.

4.4.4. Network Application Layer

4.4.4.1. Data Access Protocol (DAP)

DAP is a protocol that permits remote file access and manipulation, and file transfer, in a manner independent of the I/O structure of the node's O.S. being accessed.
The files handled may be sequentials, relatives, randoms, or indexed.

DAP defines a set of messages (fig. 41) and rules governing their exchange between two cooperating processes (cooperating entities in ISO vocabulary).
It aims to minimize overhead by defaults field specifications wherever possible, and provides 'file transfer mode' eliminating control messages exchange during transfer. It performs error detection and recovery. It handles conversions, to mask differences of file organizations, record storage formats/ representation, control characters, which are done by the requesting process. Not all conversions are supported (for instance, floating point formats or word length conversions).

The two entities or processes exchanging DAP messages are :

- The user (user process) which initiates DAP commands for accessing remote file. These commands mapped into DAP messages are transmitted via a logical link to the server's node. One and only one transport connection is associated to each file accessed.

- The server (server process) which receives DAP messages via the FAL (1), performs the user's commands, returns data and status to the user.

4.4.4.2. Services provided :

- opening/ closing/ deleting files

- creation of new file

---

(1) : the File Access Listener (FAL) provides passive DAP functions. For more information about DAP or FAL, see DEC INT ch 6 and DEC DAPFS.

Figure 41 **DAP Messages**

Message	Function
Configuration	Exchanges system capability and configuration information between DAP-speaking processes. Sent immediately after a link is established, this message contains information about the operating system, the file system, protocol version, and buffering ability.
Attributes	Provides information on how data is structured in the file being accessed. The message contains information on file organization, data type, format, record attributes, record length, size, and device characteristics.
Access	Specifies the file name and type of access requested.
Control	Sends control information to a file system and establishes data streams.
Continue-Transfer	Allows recovery from errors. Used for retry, skip, and abort after an error is reported.
Acknowledge	Acknowledges access commands and control connect messages used to establish data streams.
Access Complete	Denotes termination of access.
Data	Transfers the file data over the link.
Status	Returns the status and information on error conditions.
Key Definition Attributes Extension	Specifies key definitions for indexed files.
Allocation Attributes Extension	Specifies the character of the allocation when creating or explicitly extending a file.
Summary Attributes Extension	Returns summary information about a file.
Date Time Attributes Extension	Specifies time-related information about a file.
Protection Attributes Extension	Specifies the file protection code.
Name	Sends name information when renaming a file or obtaining a directory listing.

- read/ write of records
- error notification and correction
- format conversions

4.4.4.3. Dialoque exemple (fig. 42) :

Once a logical link (transport connection) is established, the both DAP modules (entities) exchange Configuration messages to tell which protocol version is in use, which O.S. and file system are used, which buffer size is allowed and so on.

After succesful exchange, Attributes messages supply all needed informations about the file (data type, formats, devices characteristics, access type, character set, block size, ...).

Then the Access message is sent to specify the type of operation to perform (open, create, submit a file job, modify, ...) and the file name.

Once it is done, a Control message initiates the data stream. It can specify a particular record, a range of records, or all the file.

The transfer is terminated by an Access Complete message.

Status messages are used to report on errors or processing.

4.4.5. Network Service Layer and Session Control Layer

4.4.5.1. Network Service Protocol (NSP)

The NSP is a set of messages and rules governing their exchages between two Network services modules (or entities) on behalf of both Session-entities or N.S.-entities.

The NSP is responsible for establishing, maintaining, and terminating logical links (full-

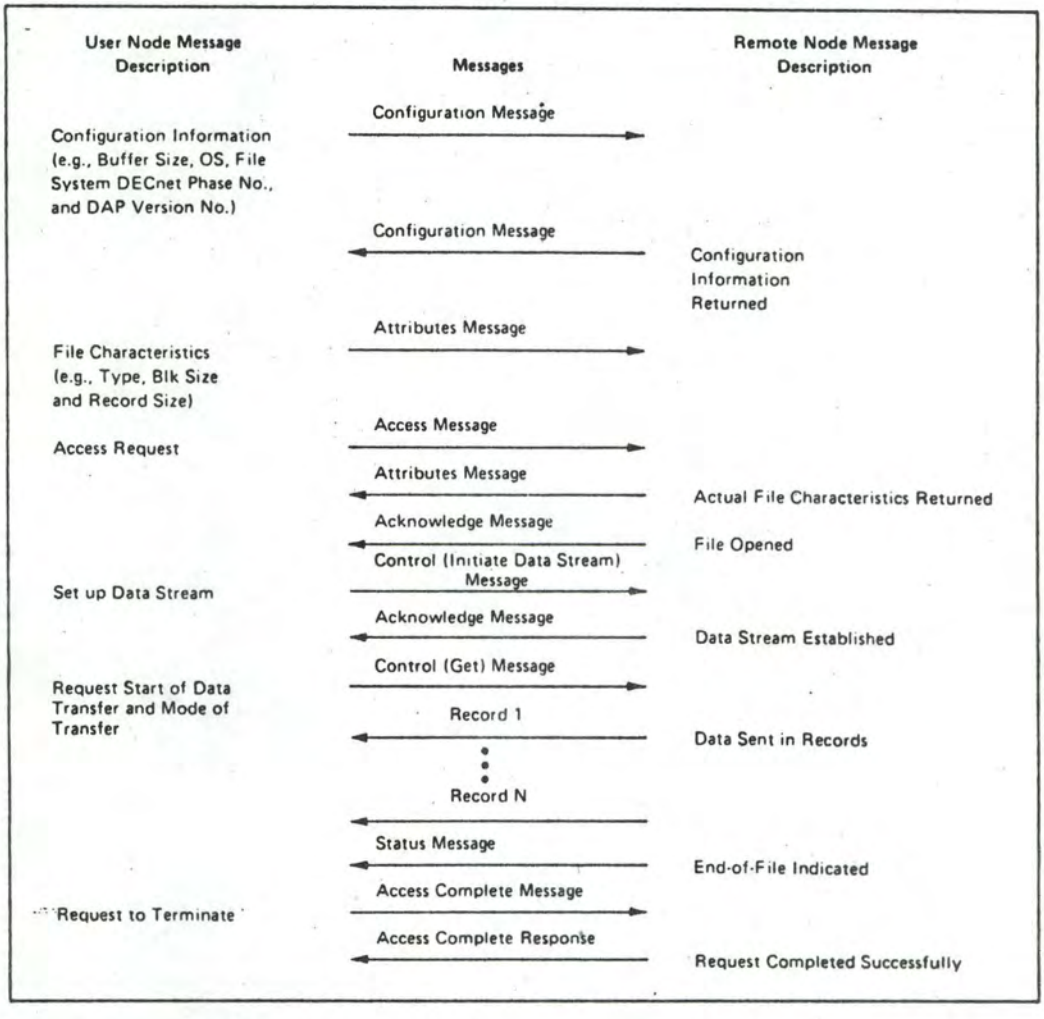


Figure 42 : DAP Message Exchange (Sequential File Retrieval)

duplex transport connections in ISO vocabulary), between Session-entities. It controls the data exchange between thes entities (sequencing, flow control, error control, segmenting, ...)

- note on logical links :

  There can be several logical links at any time between the same two NSP entities. A routing-node (phase III) can establish a logical link with any other routing-node in the network. A phase-II node can establish logical links only with adjacent nodes. Each logical link is separated from the others.

  NSP allows multiple links to share a single communication line (upward multiplexing) provides Session-entities with transport-end-point identifier mechanism called port by Digital.

  A port is an area , in memory, that contains control variables for managing the link it is associated to. The ports are managed by NSP module. A logical link has one port at each end. This port permits to the Session Control Module (session-entity) to identify and use the associated logical link.

  Each node's NSP module has a number of available ports it uses to create logical links with another NSP module (fig. 43).

4.4.5.2. Services provided.

- creation/ disconnection/ abortion of logical links
- delivery of data and control messages in sequence and free of error
- flow control
- segmenting

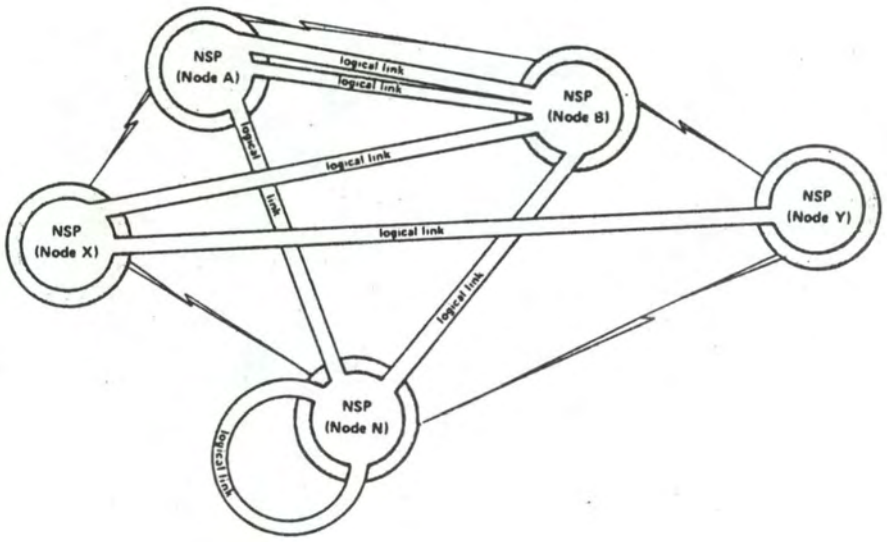


Figure 43 : Logical Links

4.4.5.3. NSP messages types :

The messages exchanged between the two Network Services modules (entities) are :

- Data
- Acknowledgment
- Control

Their meaning is summed up in figure 44.

4.4.5.4. Operation of a logical link and dialogue exemple (fig. 45)

4.4.5.4.1. Establishment of a link :

This requires the handshake process service to be runned by correspondant NSP modules (entities), in order to establish the link on common basis.

The user executes a Connect Request, providing as parameters the destination node's address, the process name in the destination node's address, the sender's identity, the connection identifier to be used for outgoing traffic, a buffer for returned data, and few other items. The Connect Request causes an NSP message Connect Initiate to be sent to the remote process.

A process (session-entity) that wants to listen for incoming Connect Initiate message must do a Receive Connect, providing a buffer for the incoming message. Once a Connect Initiate has arrived, the user may accept or reject it using Accept Connect or Reject Connect, respectively, both of which can provide return information to the other process.

If the connection is accepted, the accepting process provides its own connection identifier for outgoing traffic. NSP uses Connect Confirm and Disconnect Initiate messages to accept and reject incoming requests. However, since the network layer provides only datagram service and may suffer from old duplicates, the initiating transport station acknowledges the response.

Figure 44 **NSP Messages**

Type	Message	Description
Data	Data Segment	Carries a portion of a Session Control message. (This has been passed to Session Control from higher DNA layers and Session Control has added its own control information, if any.)
Data (also called *Other Data*)	Interrupt	Carries urgent data, originating from higher DNA layers.
	Data Request	Carries data flow control information (also called Link Service message).
	Interrupt Request	Carries interrupt flow control information (also called Link Service message).
Acknowledgment	Data Acknowledgment	Acknowledges receipt of either a Connect Confirm message or one or more Data Segment messages.
	Other Data Acknowledgment	Acknowledges receipt of one or more Interrupt, Data Request or Interrupt Request messages.
	Connect Acknowledgment	Acknowledges receipt of a Connect Initiate message.
Control	Connect Initiate	Carries a logical link connect request from a Session Control module.
	Connect Confirm	Carries a logical link connect acceptance from a Session Control module.
	Disconnect Initiate	Carries a logical link connect rejection or disconnect request from a Session Control module.
	No Resources	Sent when a Connect Initiate message is received and there are no resources to establish a new logical link (also called Disconnect Confirm message).
	Disconnect Complete	Acknowledges the receipt of a Disconnect Initiate message (also called Disconnect Confirm message).
	No Link	Sent when a message is received for a nonexisting logical link (also called Disconnect Confirm message).
	No Operation	Does nothing (included for compatibility with NSP V3.1).

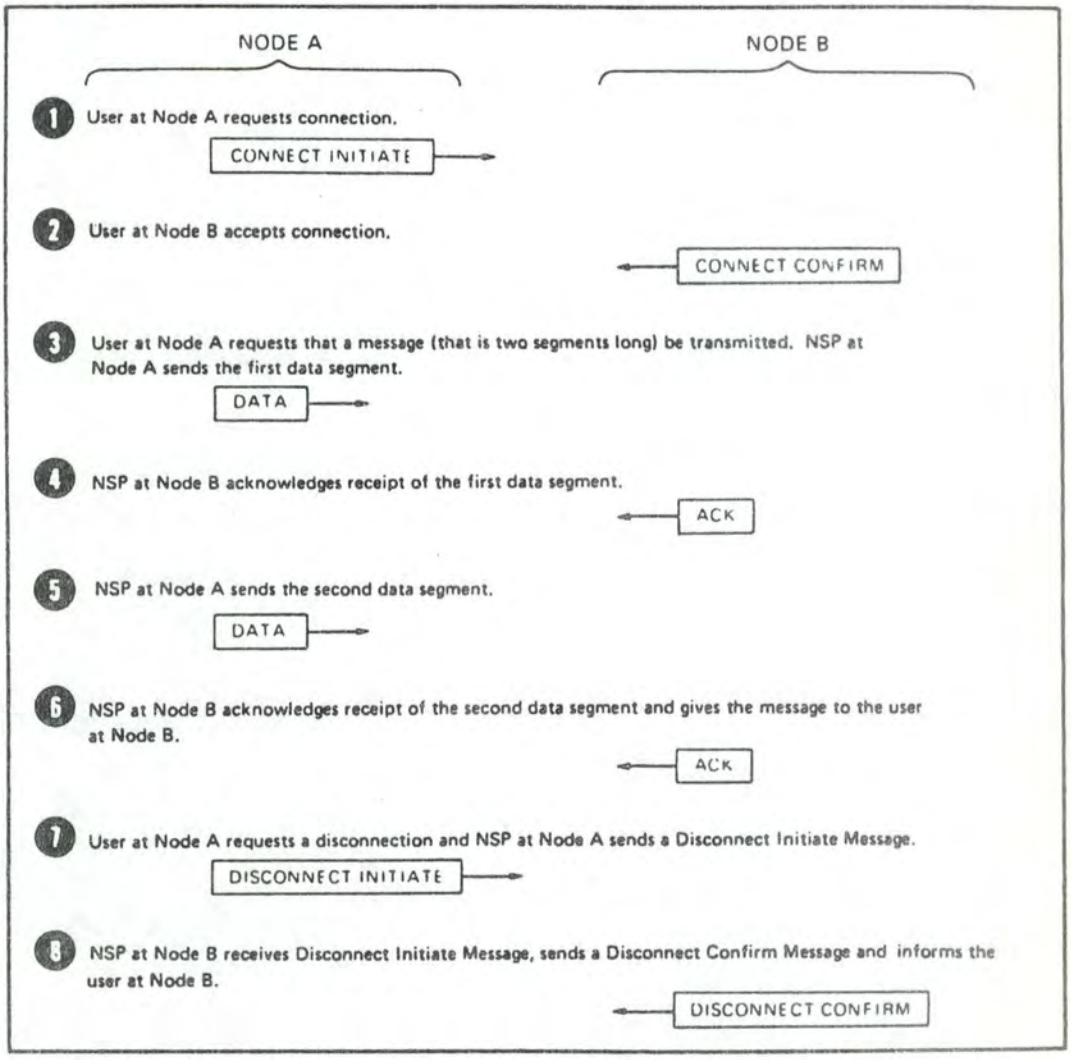


Figure 45: Typical Message Exchange Between Two Implementations of NSP

it is send, the flow control mode selectioned is exercised.

- c) Interrupt Data Flow Control

This mechanism is similar in operation to the session control message flow control basis. Meanwhile, at logical link establishment, there is an implicit request of one Interrupt message.

4.4.5.4.3. Disconnection, abortion of a logical link

Disconnection of the logical link may be initiated by either one of the end-users via the session modules connected by this logical link.

1. A Disconnect Call (Disconnect-XMT) from the session module (session-entity for ISO) results in a Disconnect Initiate sent by the NSP module. It is an orderly fashion to terminate a session and close a logical link. It ensures no data is lost, because all pending data to transmit/receive are transmitted/received before the Disconnection process completion.

2. An Abortion Call (Abort-XMT) from the session module forces the NSP module to close imediately, wether or not the remote NSP module agrees on, with a possible lost af data.

4.4.6. Transport Layer (OSI network layer)

For more complete information on Session Control operation, look at DEC SCFS.

4.4.6.1. General definitions

1. Node address and Node name :

Transport identifies nodes in a network by unique addresses; end-users identify network nodes by names (names that can be different for a same node). This implies a mapping from name to address to be done. This mapping is handled by Session layer.

For correct network operation the following rules apply :

- Transport knows nodes only by their addresses.
- all addresses are global and unique throughout the network.
- names are assigned individually in each node and are unique within a node.
- node-names - node-addresses relations are many-to-one.

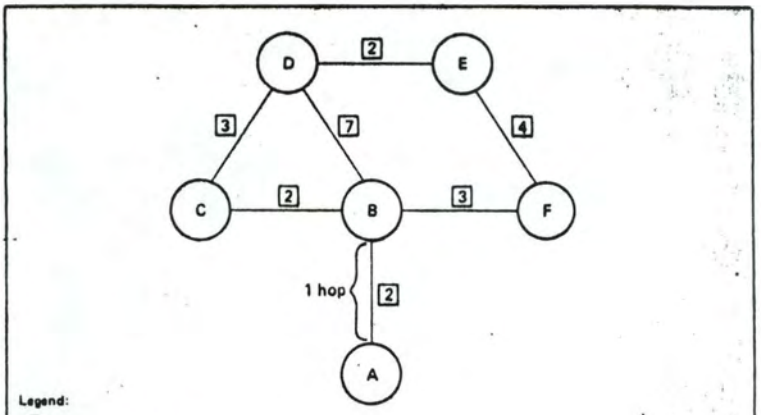
2. Path length :

> "Path [DEC GD] is the route a packet takes from source node to destination node. This can be a sequence of connected nodes between two nodes."

Path length is equal to the number of node-to-node connections through which a packet must pass to reach its destination. For exemple in figure 46, the path from A to D through node B has a length of 3.

3. Path cost :

> "Path cost [DEC DTP p 109] is the sum of the arbitrary values



Node A wants to send a packet to Node D. There are three possible paths.

PATH	PATH COST	PATH LENGTH
A to B, B to C, C to D	2 + 2 + 3 = 7*	3 hops
A to B, B to D	2 + 7 = 9	2 hops
A to B, B to F, F to E, E to D	2 + 3 + 5 + 2 = 12	4 hops

*7 is the lowest path cost; Node A therefore routes the packet to Node D via this path.

Figure 46 :Routing Terms

assigned to the physical lines that compose the path. These values are assigned by a network manager to reflect real line costs and/or arbitrary characteristics that will influence how much the line will be used."

4.4.6.2. Routing Protocol

1. It is a set of messages and rules governing their exchange between Transport entities plus a set of functions and processes contribuing to its operation.

Routing [DNA DTP] is adaptative (i.e. the system selects the best path among all possible routes) and dynamic (i.e. as status or characteristics of path in the network change, the system automatically and imediately updates the routing alternatives available). The actual path by which any message travels from its source to its destination is primarly determined by the two factor : path length and path cost.

The transport entity of each routing node maintains data bases that contain the result of algorithms that determine the path length and path cost to every possible destination in the network via every physical line emanating from that node, and designate the least costly path to each destination. Data that is used to set limits on the number and length of messages queuing up for transmission on specific physical lines to prevent bottlenecks, or to perform other types of network control, is also stored.

When a message is to be routed, the transport entity checks the data bases for the best avialable route to the requested destination and sends the message over the appropriate physical line to the next node along that path. The process is repeated at each node on the path, using that node's local data bases, until the

destination is reached.

Whenever any factor in the routing algorithm changes, such as when a physical line goes down or an operator changes the value assignated tp a line, the local node automatically recalculates the algorithms and passes the contents of its updated data bases to the other adjacent routing node in the network.

2. Services provided

- packets receiving/sending from/to any other routing node as well as an adjacent phase-II node or non routing node.

- packets routing (switching) from other nodes to other nodes. This process implies selection of the best route, updating of data bases routing information, packet forwarding, control information and path managing to be executed.

4.4.7. Data Link Control

4.4.7.1. Composition

There are two DNA protocols concerned with the Data Link layer :

- Digital Data Communication Message Protocol (DDCMP) which provide data transmision , sequencing of data , and error control to ensure data integrity .

- Maintenance Operation Protocol (MOP) which operates within DDCMP and provides network maintenance functions .

4.4.7.2. DDCMP :

4.4.7.2.1. Description :

It is a byte oriented protocol similar to the BSC of IBM or to HDLC. It relies on specific messages fields for the placement of control information , and is therefore data transparent . It avoids the overhead due to bit stuffing in bit oriented protocols . But because of the fixed block lengths and acknowledgment of DDCMP messages some overhead is also generated .

DDCMP supports half or full-duplex mode and either point-to-point or multipoint lines .

- controls the operation of the physical link between nodes and maintains the integrity and sequentiality of transmitted data .

- DDCMP groups data to be transmitted into blocks of fixed length which , with all control information appended, constitute complete messages/frames.
  The maximum message/frame length is determinated by the user and once determinated, remains constant. This size cannot overcome 16383 bytes. (since there is no preemption, if a long message is being sent over a slow line, all other traffic will have to wait [Tanenbaum 81]).

  This choice depends on several criterias (reliability and quality of transmission line, response time, type of processing, buffer size, ...)

- Sequencing of data messages is assumed to ensure proper sequencing at receiving node and to avoid lost of messages/blocks.

  The receiving node acknowledges the correct recept of data messages by returning the sequence number in response.

  A window size of up to 255 messages/frames may be selected for the transmission purpose (but sending large number of messages without requiring an ACK need highly reliable communication lines. An ACK or a NACK implies positive receipt of messages whose sequence number is less than those specified in the ACK/NACK (like in HDLC). In case of error, retransmission is assumed.

ACK/NACK can be included in data messages, or in special control messages.

- Data integrity is assured through the use of a Cyclic Redundancy Check (two bytes long). One CRC field after the header and control information and another at the end of the data text.

- DDCMP also uses time-outs and control messages to recovery from errors.

4.4.7.2.2. DDCMP messages types :

3. Data message format :

For messages received from the Transport layer to be sent across the physical link. This format ensures proper handling and error checking of both the header and the data.

- Format :

The format used for data messages/packets is shown in fig. 4 7.

The SOH field indicates start of header and indicates that a data message follows.

The Count field gives the length of the data field in bytes. A DDCMP message must be an integral number of 8-bit bytes because of the byte protocol basis.

The Flag field contains the two following flags :

- The Sync which is used to synchronize both sides of the line if necessary;

- The Select used on half-duplex or multidrop links to indicate that the sender has no more data to transfer. This allows the other side to start transmitting.

The ACK field contains the number of the last message correctly received.

The Seq field contains the current message sequence number.

F47
F48

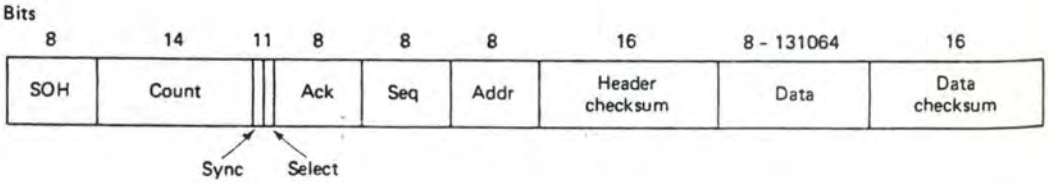


Fig. 47 : Data format used by DDCMP.

Acknowledge Message (ACK) Format

ENQ	ACKTYPE	ACKSUB	FLAGS	RESP	FILL	ADDR	BLKCK3
8	8	6	2	8	8	8	16

Negative Acknowledge Message (NAK) Format

ENQ	NAKTYPE	REASON	FLAGS	RESP	FILL	ADDR	BLKCK3

Reply to Message Number (REP) Format

ENQ	REPTYPE	REPSUB	FLAGS	FILL	NUM	ADDR	BLKCK3

Start Message (STRT) Format

ENQ	STRTTYPE	STRTSUB	FLAGS	FILL	FILL	ADDR	BLKCK3

Start Acknowledge Message (STACK) Format

ENQ	STCKTYPE	STCKSUB	FLAGS	FILL	FILL	ADDR	BLKCK3

ENQ = the control message identifier
ACKTYPE = the ACK message type with a value of 1
NAKTYPE = the NAK message type with a value of 2
REPTYPE = the REP message type with a value of 3
STRTTYPE = the STRT message type with a value of 6
STCKTYPE = the STACK message type with a value of 7
ACKSUB = the ACK subtype with a value of 0
REASON = the NAK error reason
REPSUB = the REP subtype with a value of 0
STRTSUB = the STRT subtype with a value of 0
STCKSUB = the STACK subtype with a value of 0
FLAGS = the link flags
RESP = the response number used to acknowledge received messages that checked out to be correct
FILL = a fill byte with a value of 0
ADDR = the tributary address field
BLKCK3 = the control message block check

Figure 48 : DDCMP Control Message Formats

The Addr field is used on multidrop links to specify to which station the message is addressed.

The Header CRC for control of the header. The reason for a separete CRC for the header is the sensitivity of count-delimited protocols to error. If a message [Tanebaum 81] whose true length was a few bytes was misreceived with a count of thousands, the receiver would continue listening even after the message fully arrived. This disfunction could cause protocol errors and lead to the shuting down of the line by the sender falsing concluding that the receiver is down.

The Data field contains the upper-user data .

The Data CRC is the control field for the data .

4. Control message format :

These are unumbered short messages used during link start up, resynchronization, or for messages positive/negative acknowledgment.

The five existing control messages are the following :

- Acknowledge message (ACK) : to acknowledge received messages which sequence number is less or equal to the one carried.

- Negative Acknowledge message (NAK) : acknowledges receipt of all previously transmitted messages with a sequence number less than the current message number received, and notifies the sender of error conditions related to the current message.

- Reply to Message number (REP) : to request the receiver to send back ACK or NAK concerning the last sent message. It is often used after a transmitter time-out to receive an ACK or a NAK. It is used because with such long message permitted, automatic retransmission would be highly undesirable if the receiver were merely a little bit late in sending an acknowledgment.

- Start message (STRT) : establishes, during link start-up or reinitialization, initial contact and synchronization on a DDCMP link.

- Start Acknowledge message (STACK) : response to a STRT meaning completion of initialization.

- Format : see figure 48.

5. Maintenance messages :

envelope for messages used by the MOP. They are handled by DDCMP for transmission.

- Format : see figure 49.

4.4.7.2.3. DDCMP operation :

The DDCMP module has the three following components :

- Framing : performs byte synchronization, localisation of beginning and end of received DDCMP messages.

- Link management : controls [DED GD] transmission and reception on links connected to two or more transmitters and/or receivers in a given direction. it controls the direction of data flow on half-duplex links and selection of secondary/tributary stations on multipoint links (see figure 35).

- message exchange : transfers the data correctly and in sequence over the link. It accomplishes error, sequencing and flow control using control messages exchange.

4.4.7.3. MOP (1) :

1. MOP performs the following functions :

- downline loading or upline dumping the memory of a remote node.

Maintenance Message Format

DLE	COUNT	FLAGS	FILL	FILL	ADDR	BLKCK1	DATA	BLKCK2
8	14	2	8	8	8	16	8n	16

DLE = the maintenance message identifier
COUNT = the byte count field
FLAGS = the link flags
FILL = a fill byte with a value of 0
ADDR = the tributary address field
BLKCK1 = the header block check on fields DLE through ADDR
DATA = the data field (Section 3.5)
BLKCK2 = the block check on the DATA field

Figure 49. DDCMP Maintenance Message Format

- loopback testing the data link and/or its hardware capabilities.

- restarting a remote system.

MOP is used to perform Network management functions and resources management.

2. MOP message types :

See figure 50.

4.4.8. Physical Layer

The protocols defined at this level depend on the hardware (modems, ...) and lines used, and are beyond the scope of this paper. So they are not described herein.

Figure 50: **MOP Messages**

Message	Description
Memory Load with Transfer Address (Deposit Memory and Transfer)	Causes the contents of the image data to be loaded into memory at the load address, and the system to be started at the transfer address.
Memory Load without Transfer Address (Deposit Memory)	Causes the contents of the image data to be loaded into memory at the load address.
Request Memory Dump (Examine Memory)	Requests a dump of a portion of memory to be returned in a memory dump data message.
Enter MOP Mode	Causes a system not in the MOP mode to enter MOP mode if the password matches. Usually transfers control of the satellite to a MOP program. Used for unattended satellite systems.
Request Program	Requests a program to be sent in some unspecified number of memory load messages.
Request Memory Load	Requests the next load in a loading sequence and provides error status on the previous load.
MOP Mode Running	Indicates to a host that the system is in the MOP mode and supports the features indicated in the message.
Memory Dump Data	Returns the requested memory image in response to a Request Memory Dump message.
Parameter Load with Transfer Address	Loads system parameters and transfers control to the loaded program.
Loopback Test	Tests a link by echoing the message sent by the host.
Looped Data	Returns the test message data in response to a Loopback Test message. Returned by the passive side if the message is looped from a computer.

5. DSA DESCRIPTION

5.1. Remark :

As said above in preliminary introduction, no sufficient information has been made available from CII-HB. Due partially to the fact that existing DSA descriptions are being revised.

5.2. Introduction to DSA

Distributed System Environment (DSE) is CII Honeywell's master plan for distributed processing. It is a set of protocols, programs, and hardware used to create an environment including information proccessing, data management (file creation, manipulation, storage, movment in DSE) and network proccessing (network administration and management).

DSE allows customers to set up a network, linking their systems, in order to insure communications between them. DSE supports the integration of a variety of systems and communications Protocols into a single, efficient, distributed network which supports interactive, batch or time-sharing applications simultaneously.

The architecture of DSE is called Distributed System Architecture (DSA). DSA provides a universal set of rules that govern the data movement within a DSA network.

The basic structure of DSA conforms to the International Standard Organization (ISO) OSI Reference Model, to which Honeywell has expressed a firm commitment. Using this model, DSA defines the functions performed by each layer, the protocols which control the dialogue within each layer, and the interfaces between the layers.

5.3. System Cuts

The DSA network consists of nodes (computers and terminals) joined by links, as the other existing networks.

Communications between can be via private or public communication facilities, and DSA provides specific interfacing for dedicated lines, for public X21 circuit-switching networks, public packet-switching networks such as Transpac or DATEX-P. Both Datagram type and X25 packet-switching are supported, with DSA node acting as a Gateway.

The nodes can be minicomputers, network processors or terminal controllers.

Each node has a unique identifier within the network, and within a system (or node) activities (programs, terminal users, operators, ...) have local identifiers.

The network is divided in two parts by CII-HB (fig.51):

1. the Primary Network which interconnects the nodes communicating by use of DSA rules.

2. the Secondary Networks which contain terminals and/or terminal clusters connected to a node (primary node) or between them via switched or dedicated lines or even via packet-switching networks.

In terms of System Cuts, we can identify the followings: the hosts, the network processors, the satellites, and the terminals.

5.3.1. Host :

A host is data processing oriented; it is configured with a channel-attached Front-End (FE) or integrated communication processor.

It provides services both to local and terminal users,

F51
F52

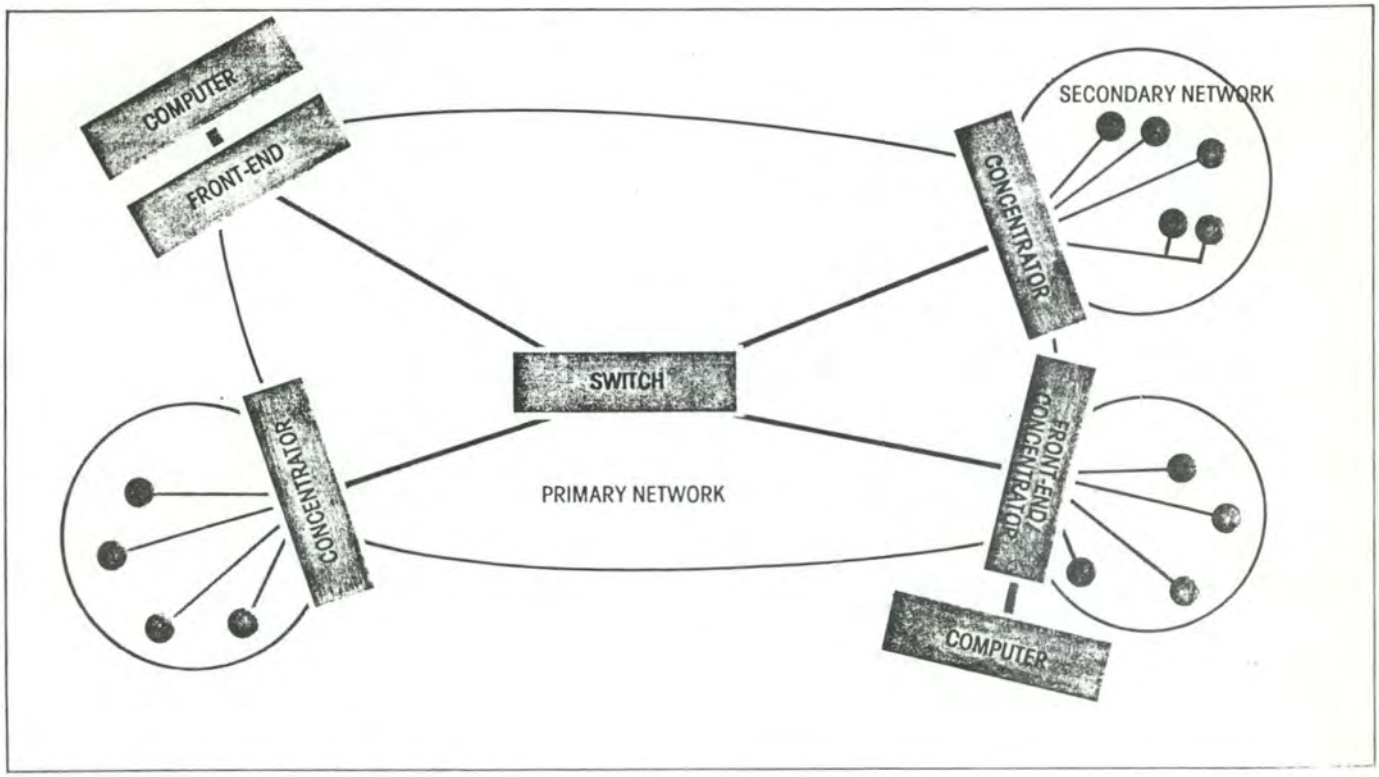


Figure 51. Network processor roles.

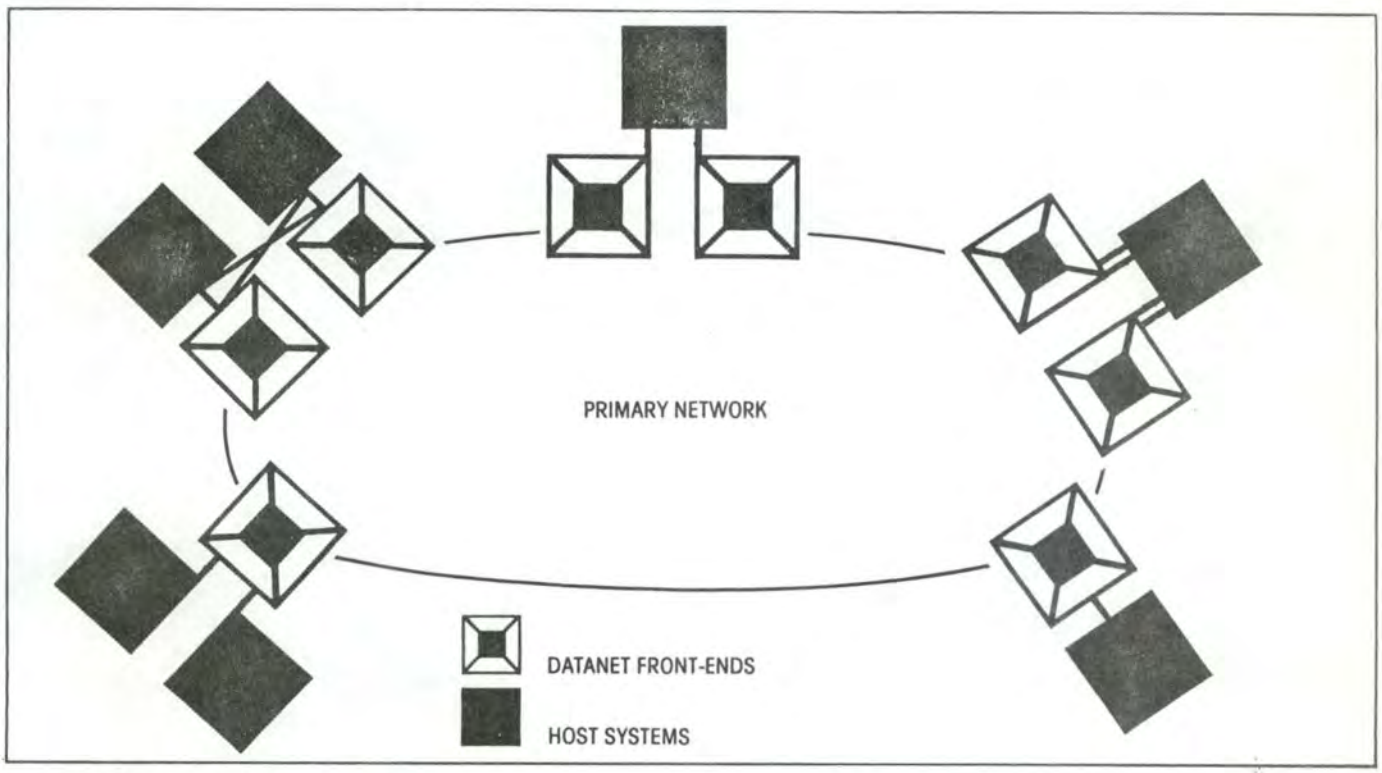


Figure 52. Front-end processor configurations.

and to other hosts or nodes in the network (the services accessed through the network transit via the FE).

CII-HB considers a host to be a large, medium or small scale computer with storage capacity. These could be DPS 8, DPS 7, 64 DPS systems of the CII-HB products.

Generally the host implements the application and message management layers (see forward sub-section 3) excepts for the pre-DSA systems (like DPS 8).

5.3.2. Network Processors :

These are computers dedicated to communication management functions. They can perform any combination of the three following roles :

1. Front-End processing providing network services to the hosts, to which they are attached, allowing them to concentrate on data processing.
   They also serve as interface between one or more hosts and a primary or secondary network (fig.52).

   - remark : acting as a gateway for non DSA host, a network processor (FE) contains both the communication management layers and a specialized data management software interfacing betwenn the DSA and non DSA elements.

2. Concentration: for acces to DSA primary network by terminals and possibly computers, using non DSA communication techniques,located in the secondary network.

3. Switching : providing routing services and network management services in the primary network.

The relations between the hosts and network processors can be 1-1, 1-N, N-1, M-N.

5.3.3. Satellites :

A satellite system serves a number of local users (terminals, ...) in its secondary network, and communicates with hosts and other satellites via the primary network, obeying DSA rules.

It provides its local users with processing, storage facilities, access to resources, within remote systems (hosts). It support a wide range of networking applications such as file transfer, remote job entry, ... , in addition to the users defined applications. A satellite does not support data processing for non local users.

5.3.4. Terminals :

A terminal is a device directly connected to a host or a satellite, or virtually connected via either a network processor, a terminal controller, a satellite to any host in the network.

It provides its users with access to programs and/or services in any processor of the network.

5.4. Service Cuts

5.4.1. DSA layering structure

The layering decomposition of DSA is that shown in figure 53. It follows the layered architecture of the OSI Reference Model and consists of seven functional layers divided into three categories by CII-HB:

- the Application Management
- the Message Management layers
- the Communications Management layers

The network management or network administration is not, like in DNA, identified as a separate layer of the architecture.

1. Application Management layer : includes all end-user interfacing application programs and intra-system operations. Among the activities that may be resident in this layer are :
   - transaction processing
   - file transfer
   - terminal concentration
   - remote batch and job entry.

   Except for specific administrative exchange protocols, DSA specifies no rules for this layer and leaves their specifications to the user responsability.

2. Message management group (MMG): concerned with message formatting, control of logical connections and messages exchange, either across the network or between processes resident in the same computer (node).

   In case of network communications, MMG interfaces with the communication management

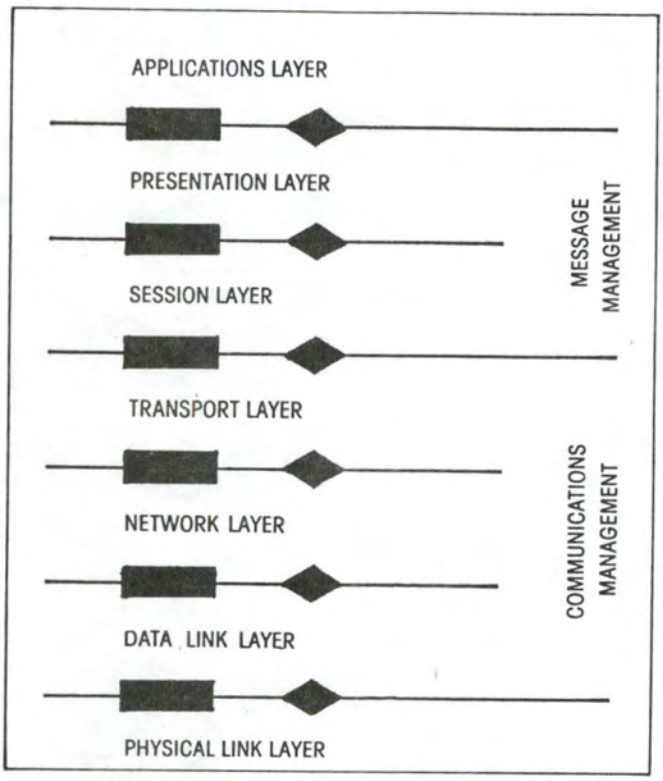


Figure 53. The DSA layers.

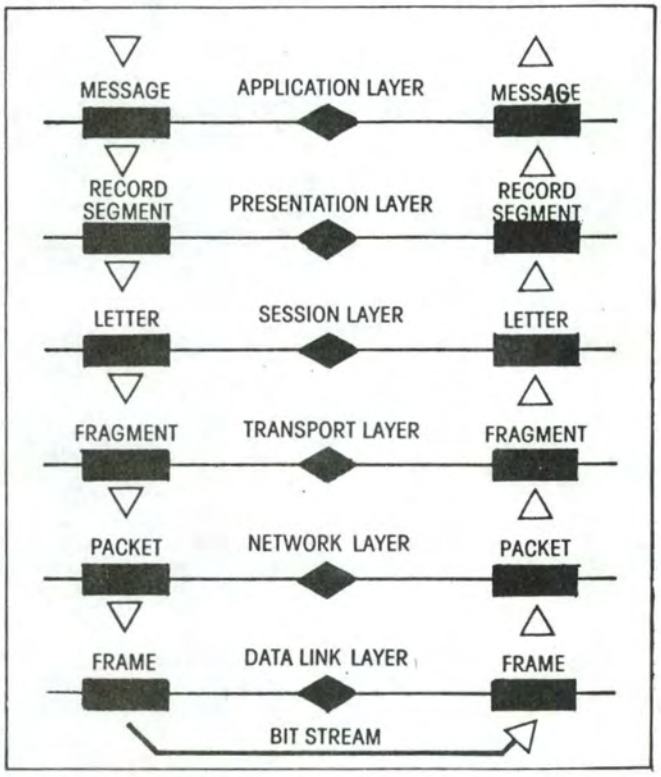


Figure 536 Transport units.

to accomplish the transfer. If the two processes reside in the same host, all functions are provided by MMG.

3. Communication management group : provides a transport service for exchanging, via the network, data between remote systems, and a total end-to-end control of the physical data communication process.

It aims to mask the nature of underlying communication medium and to assure error free data transfer.

5.4.2. Application layer

See the Application management layer above.

5.4.3. Presentation layer

5.4.3.1. Objectives

It provides the Application Layer processes with a set of services ensuring mutual comprehension. The Application processes can then dialog in a virtual mode, using a selected format.

5.4.3.2. Functions and composition

The Presentation Layer [DP DSA] reformats or transforms the data as necessary so that it can be transmitted unambiguously through subsequent DSA layers and understood by the destination processes (program or

device), regardless of the various methods used to produce the data in the Application layer and the different control codes it may contain.

Its functions can be performed by any of the three protocols resident in this layer, depending on the type of service required. These protocols, described in sub-section 4, are the Standard Device Protocol, the Transparent Protocol and Data Description Protocol.

The selection of a Protocol is done within the Session layer at connection time.

5.4.3.3. Services provided

It offers a set of data transformation services selected during the Session connection phase between the two correspondants. Data is then passed to the underlying Session layer in a standardized format masking Application Layer differences.

It also provides :

- Code conversion
- Compaction and decompaction services
- Encryption/Decryption
- Renegotiation of Presentation Protocols and options during the connection

5.4.3.4. OSI equivalents

This layer provides the following ISO OSI Presentation services :

- code conversion

- negotiation of profile
- command formatting
- data formatting, including encryption and compaction

5.4.4. Session Layer

5.4.4.1. Objectives

the Session Layer provides the upper layer and the Application processes with a dialog mechanism so that they can communicate through the network, via a Logical connection (1), regardless of the network location of the correspondants.

(1) : A Logical connection, as described by CII-HB, is a temporary or permanent end-to-end pipe (or path) between two upper layer entities (application processes) allowing them to exchange data (or letters in CII-HB terminology). It may be TWS, TWA or OW. It is physically implemented by a number of nodes and links connecting them, including the systems hosting the two corresponding entities (application processes).

5.4.4.2. Composition and functions

The functions of the Session Layer are handled by two session Protocols:

- the Connection Protocol
- the Dialog Protocol

1. Connection Protocol which performs :
   - dynamic establishment and termination of Session connections
   - negotiating common ground rules and set of choices for Presentation Layer functions (character set, code, format, compaction, encryption, transmission mode - TWS, TWA, OW

-, speed, ...)

- restart of Session connection in case of failure
- checking password and destination authentication
- requesting transport connection of the lower layer if none is available
- multiplexing several Sessions onto a single transport connection (upward multiplexing)
- mapping of the session address (mailbox) onto a transport address (plug number) to remove uses of full addressing during data exchange

2. Dialog Protocol which controls the Session once it is established and performs
   - error control to insure end-to-end data integrity and correct reception of data transmitted
   - recovery from errors by identifying restart or resynchronization points from which retransmission should begin
   - allocating buffer space for data storage
   - gathering and reporting informations about connections for network management purposes
   - segmentation
   - quarantine mechanism control
   - sequencing
   - handling Interrupt data flow and execution of associated commands

5.4.4.3. Services provided

The Presentation Layer is provided with the following services :

- Establishment of session connection on negotiation basis (negotiation and selection of one of the three Presentation protocols, and session facilities selection)

- Restart of session connection in case of physical failure

- Renegotiation of Presentation and Session parameters during the Session

- Resynchronization mechanism allowing Application processes (entities) to define points from which to begin retransmission in case of unrecoverable error

- Quarantine services : mechanism by which a correspondent controls when the receiving Session layer (Dialog Protocol) will pass data to the receiving correspondant (correspondents mean Presentation entities here)

- Normal data exchange

- Direction flow control by the mechanism of interaction units which determines the change of turn on TWA Session connections

- Interrupt data exchange which allows Presentation entities to exchange short proritary messages/coomands influencing the session and Normal data exchange (demand turn change, interrupt request, purge request for a current quarentine unit (2))

5.4.4.4. OSI equivalents

This layer globally provides the following services of the ISO OSI Session layer :

- Session establishment and session termination

- Context management

---

(2) : A quarantine unit consists of one or more record segments accumulated by the receiving Session entity until an end of quarantine unit is detected. See Protocol cuts for more informations.

- Session identification
- Data exchange of session-service-data units
- Data delimiting :
  - › quarantine unit mechanism
  - › session interaction unit mechanism (turn change, recovery rights,
- Dialogue management (TWS, TWA, OW)

5.4.5. Transport Layer

5.4.5.1. Objectives

The basic objective of this layer is to move data between the two end connected Applications/systems without introducing any error and to notify Session control if this becomes impossible. It masks the physical nature of the communication medium to the upper layers and makes optimum use of the available communication resources (considering response time, costs, lines sharing, ...).

5.4.5.2. Functions

To realize these objectives the Transport Layer performs the following functions :

1. At establishment phase
   - Identification of the correspondant by its network plug number (network address) composed of the system number (unique and meaningful within the network) and the local plug number (meaningful within the addressed system).
   - Mapping transport addresses onto network addresses.

- Multiplexing several transport connections onto a single network connection (upward multiplexing) if necessary.

- Selection of the best network service.

- Option negotiation : the two upper Session entities agree on a set of functions the transport layer will ensure during data exchange phase :

  - Transport data unit maximum size (fragments)

  - Type of error control:

    › Data integrity checks : using redundancy checking mechanism.

    › Sequentiality checking : to insure that blocks of data are received in a correct order, and to detect loss or duplication of messages.

    › Time out : to ensure that the loss of data blocks will not remain undetected, and permit recovery from situations where one of the cooperating processes fails.

  - Type of flow control and credit value.

  - Which side is responsible for recovery.

2. During data exchange phase :

- Fragmentation (and reassembling) if necessary, of the letters (session data units) received from the session layer to respect network constraints and insure better error checking.

- Error checking and recovery by sequencing of the fragments (modulo 128) to insure no lost and no duplication. This is not always necessary if yet performed by the lower network layer.

- End-to-end flow control, by a mechanism of credit, on each transport connection. The credit value can be dynamically adjusted by the corresponding transport layers, taking account of network congestion and other parameters, in order to take the fullest possible advantage of the available

communication resources.

3. Termination phase :

- orderly termination of the transport connection.

5.4.5.3. Services provided

The services provided to the Session layer are :

- Transport connection request and confirm
- Disconnection request and confirm
- Data transfer service free of error
- Abortion of transport connection

negotiation of transport connection parameters during the establishment phase.

5.4.5.4. OSI equivalents

The OSI Reference Model Transport Layer functions performed by this layer are :

- Transport connection establishment/termination
- A part of the class of services selection process
- Data transfer functions (error control, recovery, flow control, normal data exchange, segmenting, reassembling, multiplexing, data unit size selection). Purge and expedited services seem not to be provided.
- Mapping of transport address onto network address

5.4.6. Network Layer

5.4.6.1. Objectives

The Network Layer [CII DSA] (also called the routing layer or path control layer) provides the means for two transport entities to exchange data over a network. It removes transport layer from routing or switching considerations.

It sets up and manages a route that may pass through various data links, network nodes and (external/public) networks, creating a physical path by which corresponding transport entities may exchange data [DP DSA].

- Remark :

  The Datagram service specified in the following may be no more present in the current DSA releases.

5.4.6.2. Functions

In order to realize these routing objectives, the Network layer performs the following functions :

1. Network connection establishment :

   This is a switching function between any of the four link types that may be used :

   a) Dedicated point-to-point communication lines:

   - mapping of the connection onto the appropriate data link.

   b) X21 circuit switching path :

   - automatic calling procedure to establish the connection with the remote system.

   c) Datagram path through an X25 packect-switching network :

   - requires no connection establishment since each packet contains full source and destination addresses.

   D) Virtual circuit path through an X25

packet-switching network :

- creation of the virtual circuit is done with a virtual call following the X25 protocol. This circuit can be permanent or switched dynamically on a call by call basis through the network.

2. Data transfer :

Following the medium selected, the network layer provides or not function that will not have/or have to be accomplished by the transport layer.

a) dedicated lines : no own control functions

b) X21 circuit switching :

- no flow control or error control
- multiplexing of several switched circuits onto a simple data link connection

c) Datagram X25 : depends on the implementation

d) Virtual circuit X25 :

- end-to-end flow control
- multiplexing of V.C. onto a simple physical path
- error checking for lost of packets on a V.C.
- interrupt and normal data packets transfer

3. Termination of the network connection : performed depending on the used medium.

5.4.6.3. Services provided

The services provided depend on the medium

selected and are not detailed here.

Globally we can identify :

- network connection establishment/termination
- data exchange
- specific services

5.4.6.4. OSI equivalents (3)

Depends on the implementation;

- The dedicated lines and Datagram provides the non-connection oriented sevices of the OSI Reference Model Network Layer (Data Request, Data indication, maximum length of the packet).
- X21 and X25 provide the connection oriented services of circuit-switched subnetwork and X25 subnetwork respectively.

5.4.7. Data Link layer

5.4.7.1. Objectives

The data link layer [CII DSA] handles the transfer of data between adjacent network nodes linked by a physical communication medium. It ensures a without loss or error transmission.

5.4.7.2. Functions

The functions performed are those described in the HDLC Lap-B Protocol (1).

---

(3) : look at ECMA/TC24/81/78

(1) : look at Standard ECMA-61, August 79, Balanced class of Procedure, for more information.

The Lap-B designs the Balanced Procedure. Either of the two nodes can initiate transfer (no master/slave relationships), control transmission initialization and recovery procedures.

- Remark : the communications within a secondary network are not necessarly governed by DSA data link HDLC procedures but generally utilizes another protocol such as VIP, TTY, ...

5.4.7.3. Services provided

Essentially the same that the ones provided by the HDLC procedures. For more informations refer to note (1).

5.4.7.4. OSI equivalents

They are the following of the data link layer :

- Data link connection establishment/termination
- data exchange
- sequencing
- error notification or recovery
- flow control

5.4.8. Physical layer

5.4.8.1. Objectives

Defines the electrical interface that allows the connection of the data equipment to data communication equipment.

5.4.8.2. Functions and Services provided

These functions are described in the existing Standards and Recommendations and are thus not described here.

5.4.9. Network Administration

5.4.9.1. Objectives

These services are provided and performed by the network processors (Datanet 7100). They are distributed among three modules : the Node Administrator (NAD) present in every network processor, the Network Administration Storage Facilities (NASF), and the Network Operator Interface (NOI).

5.4.9.2. Services provided

5.4.9.2.1. NOI :

Provides the network operators with access to the Network Administration services located in the NAD (fig. 55).

This interface controls the operator commands (priority, privileges, validity, ...), formats them conforming to the Administration Exchange Protocol, and forward them to the appropriate NAD. Similarly it receives responses from the various NADs and reflects them to the operators.

5.4.9.2.2. NAD :

Provides the administration services required in each node :

- handling and executing operator commands

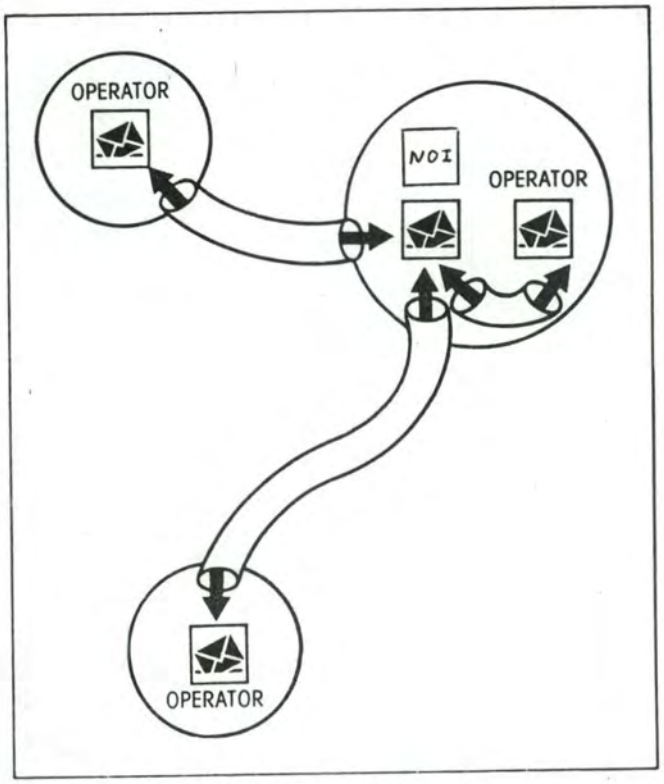


Figure 55. Network Operator Interface.

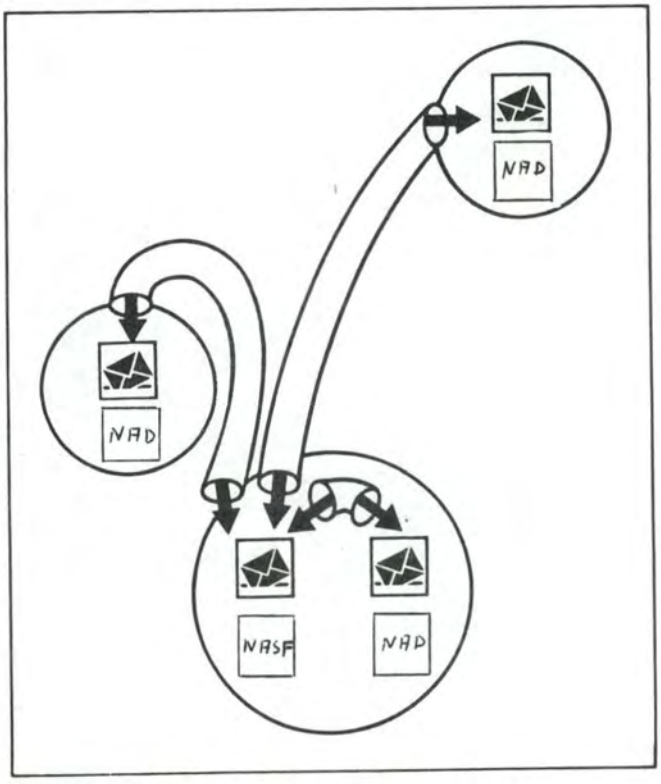


Figure 56. Network Administration Storage Facility.

- controling adjacent nodes at start-up and after failure
- timing
- diagnosis
- monitoring network operation and reporting errors
- accumulating statistics fromm all the layers and forwarding them to an NASF

5.4.9.2.3. NASF :

Provides storage for software and data concerning the network (fig. 56);

- statistics about accounting and network operation
- error information
- network and node configuration information
- access control information

The NASF is implemented on a host system and exchanges information with NADs using the Administration Exchange protocol.

5.4.9.3. OSI equivalent

These services are similars to those provided in the Application Layer by the System-management-application-processes.

5.5. Protocol cuts

5.5.1. Application layer

The Application management contains the system and user application processes of the Application layer. It is not the concern of DSA [CII DSA].
The protocols defined are thus beyond the scope of this paper.

5.5.2. Presentation Layer

5.5.2.1. Composition

The dialogue at this level is ruled by different protocols, depending on the negotiation at connection time, terminal involved, and the services required.

The selection process by which the type of protocol to be used, and other Presentation layer options, are chosen is controlled by and occurs in the Session Layer. The protocol and options may be renegotiated during a session.
The selectionable parameters are : Protocol type, compaction, encryption, character set, code conversion, data structure.

Three different Protocols are identified :

- The Standard Device Protocol
- The Transparent Protocol
- The Data Description Protocol

5.5.2.2. Standard Device Protocol

Maintains tables grouping various types of

terminals that share a common set of characteristics and makes possible for Application processes (programs, ...) to ignore the specific characteristics of terminals.

Three classes of services are allowed that serve alphanumeric and pseudo-graphic terminals and/or processes:

1. Minimum class : handles sequential devices (teleprinters, ...) and allows addressing by character position on the device.

2. Text class provides direct addressing of the device (line and row position).

3. Form class provides field addressing and handling capabilities.

Added to these three classes exists the control functions aiming to correctly assure the data exchange. They handle the translation of commands (tab, blink on, line feed, ...) to offer a standardized format of control functions for every terminal, and mask the diferences to application processes.

5.5.2.3. Transparent Protocol

This protocol only provides compaction, code conversion, connecting services, interfacing between the Application Layer and the session Layer . Data and commands are passed transparently through the Presentation Layer.

This protocol is requested for device specific applications (processes) or for non DSA environment applications running in a DSA network.

5.5.2.4. Data Description Protocol

Still under development. It would contains a data formatting service similar to the shema approach of CODASYL Data Base management.

5.5.3. Session Layer

5.5.3.1. Composition

The Session control provides services that allow two application layer processes to cooperate (via the Presentation Layer). Those services act according to protocols.

1. Connection Protocol : which rules the connection-services used to establish and release logical connections (Session connections) between activities (Application entities).

2. Dialoque Protocol : which rules the exchange of data during the data transfer phase on base of parameters negotiated at connection time.

5.5.3.2. Connection Protocol

5.5.3.2.1. Services provided and functions performed

1. Connection mechanism :

Permits one process (Presentation entity) to establish a logical connection with another by the use of an Initiate Logical Connection Request Letter containing
- the name of the system in which the called process is located,
- the name by which the called process is known within its system.

The Session control determines the Transport services to use and assigns a number to the logical connection (session connection).
The Transport system then generates the context for the Logical connection, including a plug number (transport address) to identify the Transport Service Access point associated with the mailbox of the calling user.

The called process responds with an AILCR

Letter (Acknowledge Initiate Logical Connection Request Letter) and generates its own context (plug number, ...).

2. Option negotiation

The initiating letter contains information used to negotiate session options and agree on cooperation.
These are :

- The type of flow : TWS, TWA, One Way
- selection of two-way flow of Interrupt signals independent of Normal flow, and having priority on it.
- correspondent responsible for initiating recovery
- synchronisation and quarantine mechanisms
- interaction management (turn changes, ...)
- ...

3. Termination and recovery

Handles the termination and/or recovery process on behalf of upper layer.

It includes :

- terminate/restart a session connection
- reestablishment and acknowledgment of session in progress.

5.5.3.3. Dialog Protocol

5.5.3.3.1. Message types :

1. Delimitation and synchronization (fig. 56a)

The data exchanged between corresponding entities is subdivised into records nested inside quarantine units nested inside interaction units.

a) Session Interaction Unit: a Session Interaction Unit [DSA GD] gives the correspondents explicit control over the flow of data on a two-way-alternate logical connection. The "end of interaction" flag corresponds to your turn. An interaction unit may contain one or several quarantine units. (Initial turn is one option negotiated during logical connection establishment).

b) Session Quarantine Unit : a Session Quarantine unit [DSA GD] allows a correspondent to control when the receiving dialog service passes data to the receiving correspondent. It consists of one or more records, which will be accumulated by the receiving dialog service until it detects an "end-of-quarantine unit" flag, set by the sending dialog service at the request of the sending application entity.

This implies enough buffer space to be provided on each side in order to hold a quarantine unit (buffer space amount is negotiated at connection establishment).

c) Record : a record is a logically complete data unit that an Application-layer entity passes to Session control. It may be of any size. The Session control service may fragments the record into record segments in order to respect physical constraints such as buffer sizes, in which case it marks the last segment with an end-of-record flag.

d) Interrupts : privileged commands sent as data but processed in priority by the receiving session control service.

They are used to ask the remote correspondent (entity) to:

- Execute a specific action (attention interrupt).
- Execute a specific action and hand control over to the caller (demand turn

F 56a.
b.
c.

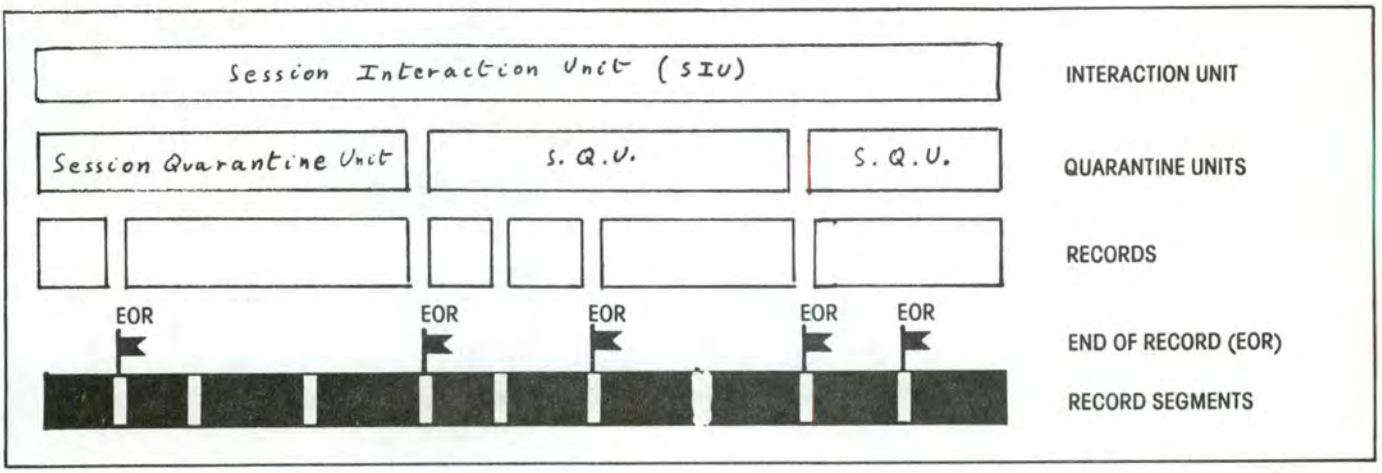


Figure 56a. Session Interaction Units and Session Quarantine Units.

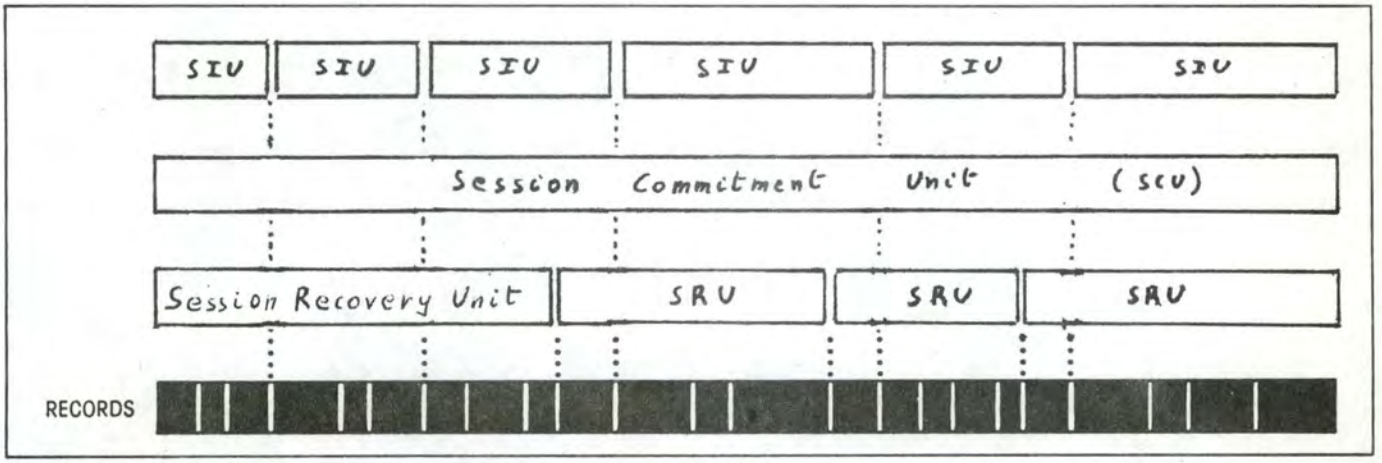


Figure 56b. Session Commitment Units and Session Recovery Units.

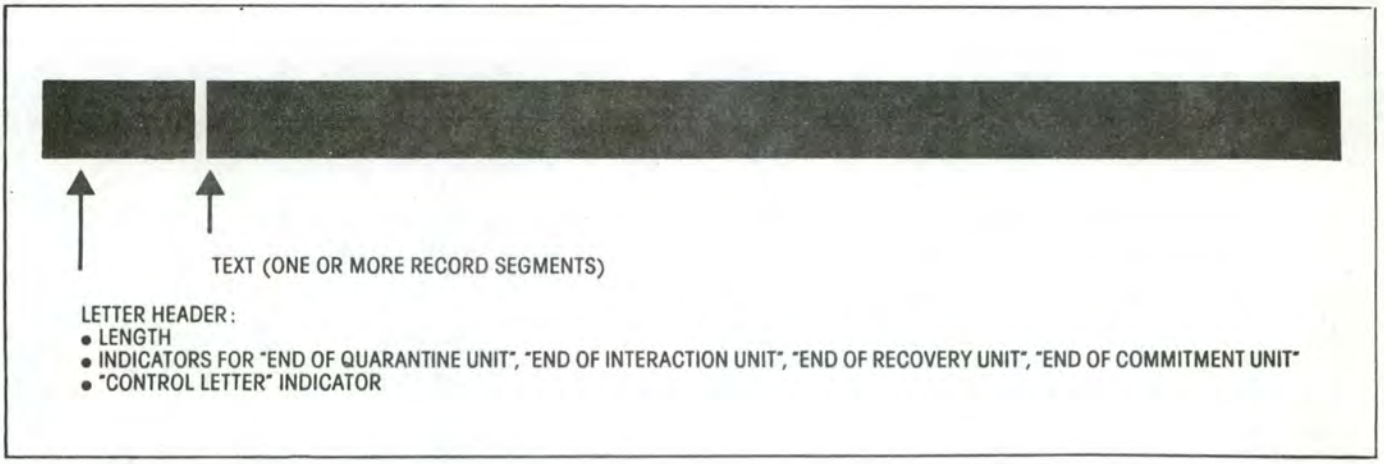


Figure 56c, Letter format.

interrupt).

- Execute a specific action, suppresssing the records of the current quarantine unit, which will not yet have been delivered to the application layer (Purge interrupt).

2. Error recovery (fig. 56b) :

a) Session Recovery Unit : the Session Recovery Unit [DSA GD] corresponds to the interval between two checkpoint. Each Application layer process specifies the point at which a recovery unit ends. Session control numbers the recovery units.

To recover lost data a correspondent can send an explicit command (Recover), using the recovery unit number to identify the point from which to begin retransmission.

b) Session Commitment Unit : a Session Commitment Unit may contain one or several recovery units, and one or several interaction units. It signals the point at which a logical operation can be assumed to be complete. (Typically it might correspond to the end of a transaction with all data-base updates sucessfully performed.) Commitments units can be numbered as for the recovery unit.

5.5.3.4. Function

- Fragmentation and blocking.
- Error recovery through the synchronization mechanism using recovery or commitment units.
- Synchronization and delimitation of data exchanged on behalf of the corresponding upper entities.

5.5.4. Transport Layer

5.5.4.1. System Communication Facility

This module performs the functions related to the Transport layer. It realizes the Transport connection initializations (requested by Session Layer), maintains and manages each connection, and handles the connection termination process.

For these tasks to be executed SCF requires the help of task management and buffer management modules. The former handles resources management and tasks activation, the latter manages logical buffers needful for data manipulation activities.

5.5.4.2. Services provided

The same as those mentioned in Service cuts :

upward multiplexing, fragmentation and reassembling, flow control (credit), error recovery, timeout management, sequencing, ...

5.5.4.3. Operation and example of a Transport connection establishment

1. Receiving a connection establishment request from session entity, the SCF

   a) allocates resources and creates a Transport Station task,

   b) sends a Transport connection Request (FLCT) to the SCF in the destination node hosting the correspondant Session entity. This request contains parameters and the plug number of the initiating Transport Station.

2. The receiving SCF creates a Transport Station task, selects a plug number to identify the connection, and establishes this latter - if possible - according to the connection parameters.

The receiving SCF sends back an Accept (FL-CC) or a Refuse (FL-DT) letter.

3. If the connection is accepted, the initiating Transport Station sends an Acknowledge letter including the Session-connection Request (ILCRL) and its parameters, and matches the options accepted by the distant transport Station.

4. The receiving transport Station passes then the ILCR Letter to the Session layer, triggers a time out and waits for the ILCR response. If the response is 'OK', the SCF sends back an acceptation letter (ILCA Letter) to the intiating SCf for the intiating Session layer.

5. On reception of the ILCA Letter starts the dialog between the two Session entities, using the transport services.

The termination process is similar.

5.5.5. Network Layer

No information available on Protocols used.

For what concers the Virtual circuit services (X25...) refers to any X25 specification for more information.

5.5.6. Data Link Layer

5.5.6.1. HDLC Lap-B Protocol

For a complete description refers to [ECMA 40,49,60,61].

5.5.7. Physical layer

The protocol defined at this level depend on the hardware and lines used, and are beyond the scope of this paper. So, they are not described herein.

6. CNA DESCRIPTION

6.1. Remark :

No sufficent information has been made available in time. This is due to the fact that CNA concepts and architecture were being revised. We appologize thus for this description being uncomplete.

6.2. Introduction to CNA

Communication Network Architecture (CNA) is the network architecture developed by NCR. "CNA is a direct result of strategy (business plan). CNA is a technical plan that ensures the connectability of products by defining the communication interrelationships among NCR products. CNA reflects the corporate business objectives in the specification of these communication interrelationships. The scope of CNA includes all those functions and equipment which enable people, processes and devices to communicate".

CNA is defined by a set of formats, protocols, programs and hardware used for networking in various environments (SNA, Packet-Switching, Circuit-Switching, BSC, Start/stop, OSI and DCNA environments).

CNA is presented as a "conceptual template, or meta-architecture, used for the primary purpose of describing how multiple, widely differing communcations architectures and technologies fit together and intercommunicate under CNA". "This meta-architecture provides a structure which defines the functional and topological bound of CNA, but also allows different environment expansions".

The structuring of CNA architecture model seems to be similar to the OSI one, even if this latter is described as a particular environment to cope with.

It is to point out that CNA includes gateway services, called Service groups, which provide a means, a conceptual mold into which the formats and protocols of disimilars environments

(systems) can be mapped.

The layered structure of CNA, its major services, and its relation to OSI layered architecture are shown in figures 57 and 58 respectively.

6.3. System cuts

No information available.

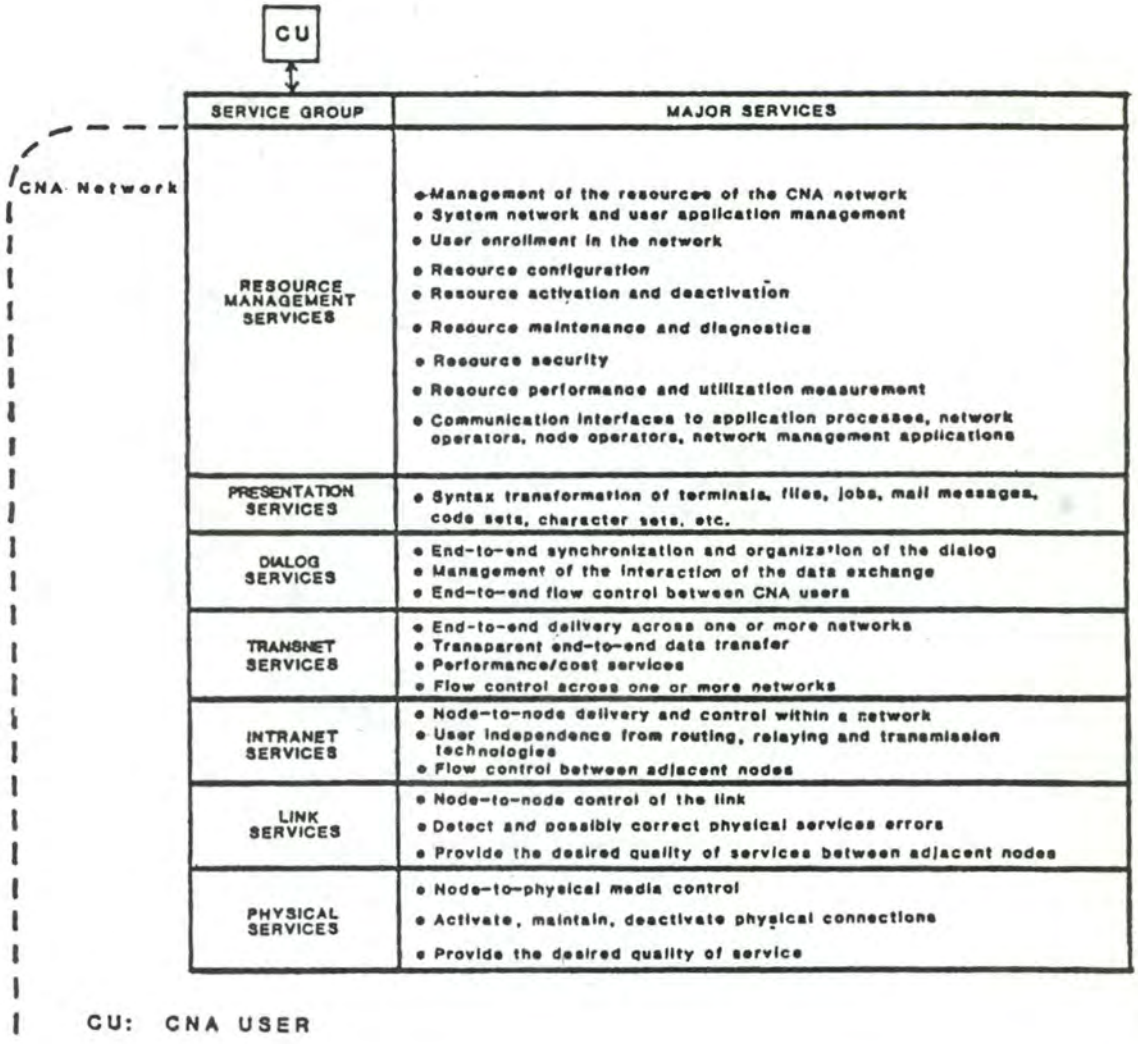


Figure 57. Summary of CNA Services

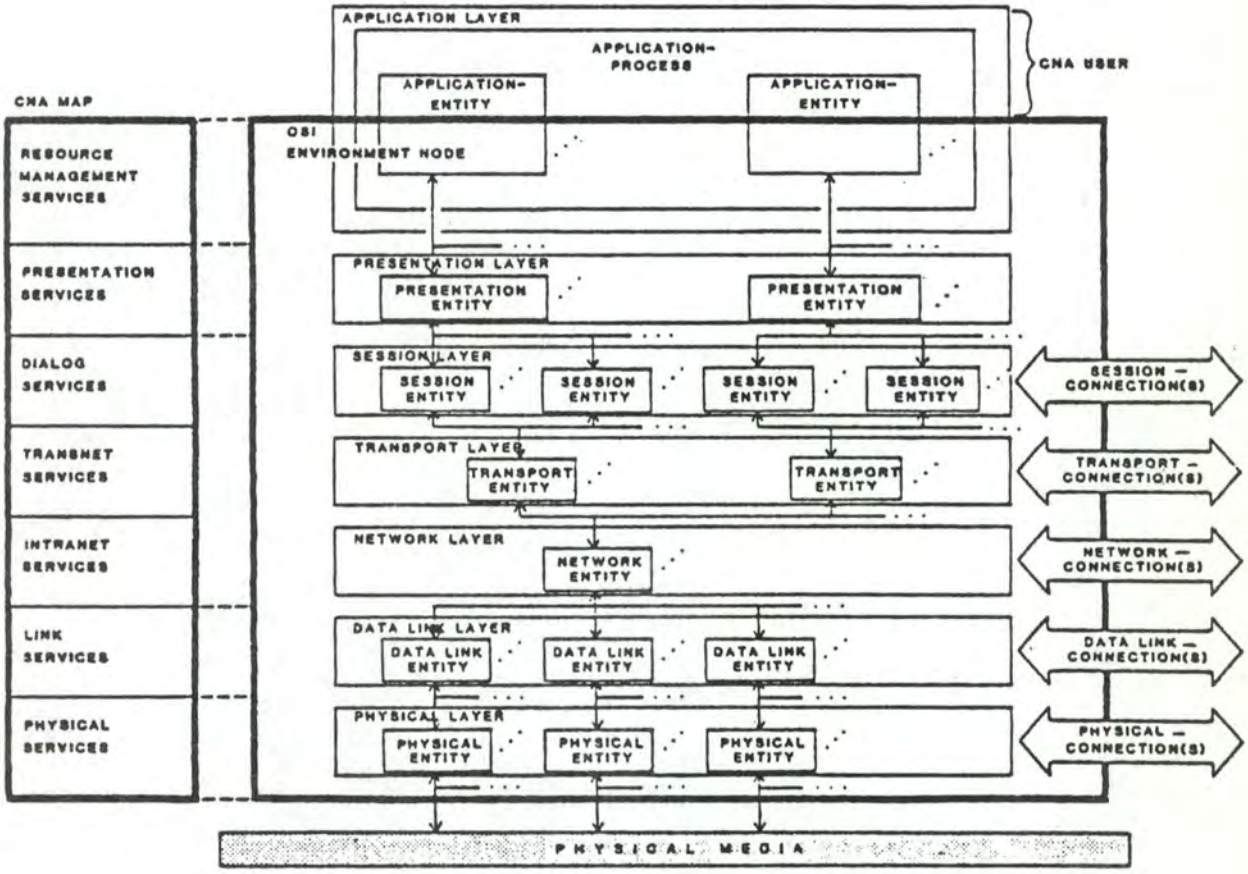


Figure 58. Overview of an OSI Environment Node and Its Relationship to CNA

6.4. Service cuts (1)

Only a short sum up of the overall purposes and services provided by the different CNA layers is presented, for the documentation was received too late.

For each of the layers identified [CNA DDR], the next sections provide:

1. The overall purpose of the layer.
2. A typical, but not necessarily exhaustive, list of services within each layer.

6.4.1. Resource Management Services

CNA resource management services provide services which direct and supervise the control of: 1) the network addressable entities (e.g., nodes, links, connections, etc.); and 2) the procedures associated with them (e.g., network directory support, dialog selection, etc.).

a) Identification of intended communication partners (by name, by address, by definite description, by generic description)
b) Determination of the current availability of the intended communicants
c) Establishment of authority to communicate
d) Agreement on privacy mechanisms required
e) Authentication of intended communicants
f) Determination of cost allocation methodology
g) Determination of the adequacy of required resources
h) Determination of acceptable quality of service (response time, tolerable error rate, cost in relation to the previous considerations)
i) Synchronization of cooperating CNA users
j) Selection of dialog discipline including initiation and release procedures
k) Agreement on responsibility for error recovery
l) Agreement on procedure for control of data integrity
m) Identification of contraints on data syntax (character sets, data structure)
n) Initiation, maintenance and termination of CNA users
o) Configuration and program loading of CNA resources
p) Allocation and de-allocation of CNA resources to CNA users
q) Maintenance of CNA resources

---

(1) : the following sections are reproduced from [CNA DDR].

r) Detection and prevention of CNA resource interference and deadlock
s) Error detection, checkpointing and recovery control for CNA users
t) Establisnment, maintenance and release of connections between entities which manage the network
u) CNA resource utilization and performance measurement
v) Processing of application program communication primitives (e.g., a COBOL read/write interface to access communication functions)
w) Processing of network operator interface languages (e.g., a network control language or network definition that is common to all NCR products)
x) Translation of protocols between diverse environments (e.g., conversion of BSC line discipline into SNA requests and responses).

6.4.2. Presentation Services

CNA presentation services provides services involving CNA supported end-to-end syntax transformations in behalf of a CNA user without altering the meaning of the data.

a) Conversion of the real syntax of terminals, files, jobs, and mail messages into an architected syntax
b) Data syntax transformation (e.g., code and character set conversions, compression and compaction)
c) Data formatting, i.e., modification of the layout of the data
d) Transformation of the image formats to be presented (e.g., display image formats)
e) Initial selection and subsequent modification of the transformation and formats to be used
f) Initial selection and subsequent modification of the definition of images to be presented,

6.4.3. Dialog Services

CNA dialog services provides services involving end-to-end control of a conversation.

a) Organization and synchronization of the data exchange between users of this service
b) Management of the data exchange interaction between users of this service
c) Control of the flow of data between the users of this service

6.4.4. Transnet Services

CNA transnet services provides services involving end-to-end delivery across one or more networks.

a) Provide transparent transfer of data between the users of this service
b) Provide end-to-end in sequence delivery to the users of this service
c) Provide end-to-end error detection and recovery relative to the quality of service required by the user of this service
d) Enable logical connections within one or more networks to be shared by multiple end-to-end connections
e) Provide each user of this service with the required performance and quality of service at a minimum cost
f) Provide control of the end-to-end flow of data across one or more networks

6.4.5. Intranet Services

CNA intranet services, as illustrated in Figure 59 , provides services involving node-to-node delivery and control within a network.

a) Provide the means to establish, maintain and terminate a connection between nodes within a network
b) Provide to the end-to-end users of this service independence from routing and relaying considerations and independence from the differences in the characteristics of different transmission technologies
c) Provide control of the flow of data between adjacent nodes to avoid congestion

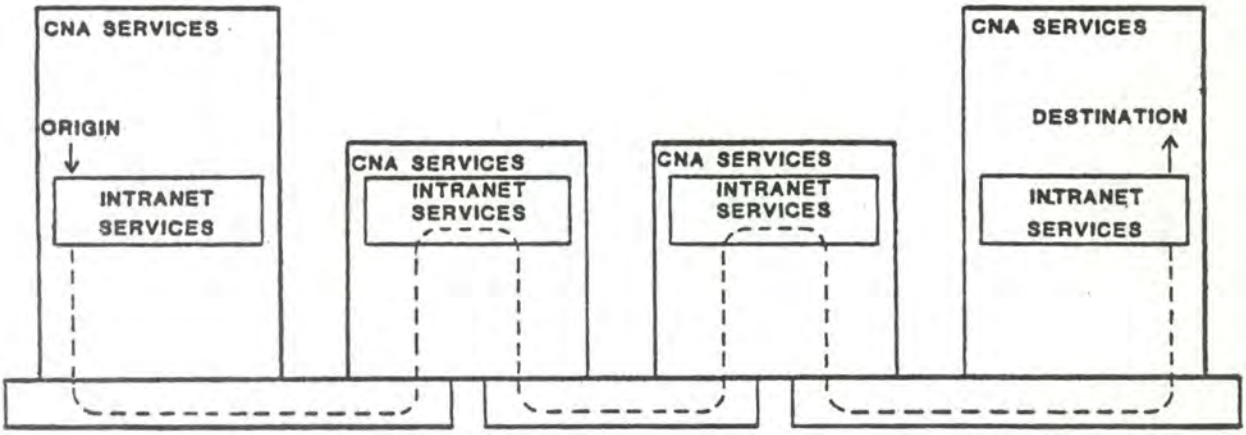


Figure 59. Intranet Services Concept

Since intranet services provides the primary node-to-node delivery or routing service of CNA, it is important for the meta-architecture to provide a structure for delineating these services within the CNA network. This structure is called the intranetwork. As illustrated in Figure 60, an intranetwork is a subnetwork of the CNA network which conceptually delineates, for each specific CNA environment, the composite intranet services, link services, and physical services components in all of the interconnected environment nodes for that environment. For instance, in the SNA environment, the composite path control, data link control and physical control components for all interconnected SNA nodes would comprise the SNA intranetwork.

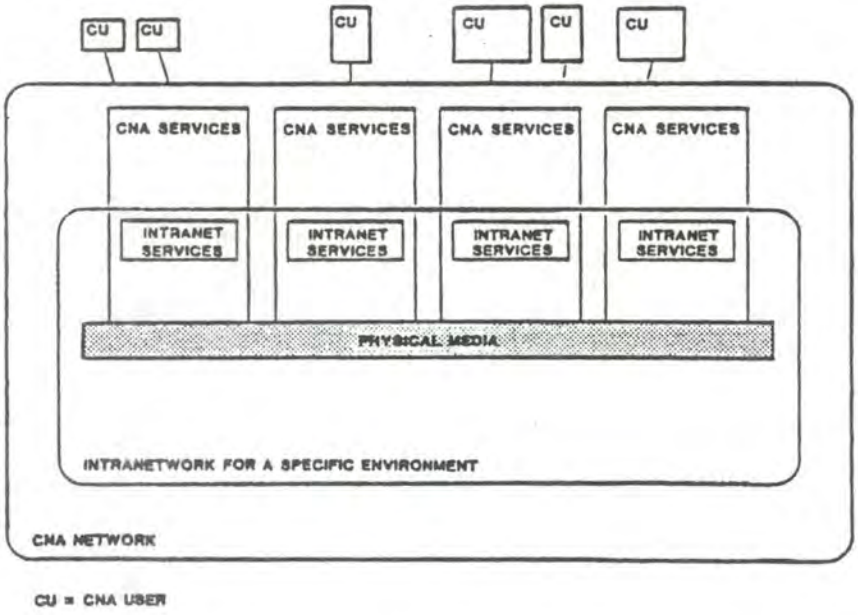


Figure 60. CNA Intranetwork Concept

6.4.6. Link Services

CNA link services, as illustrated in Figure 61 , provides services involving adjacent node-to-node control of the link.

a) Detect and possibly correct errors which may occur in the physical service. This include functions such as:
   - ensure that data is received in the order that it is transmitted
   - provide for re-transmission of data when necessary
   - gather link level diagnostic information (e.g., link level loop or echo tests)

b) Provide, when possible, the desired quality of service between adjacent nodes. This includes functions such as:
   - provide switched and non-switched connections
   - provide the desired flow control between adjacent nodes
   - allow transmit-receive direction options such as two-way alternating or two-way simultaneous

c) Enable the users of this service to control interconnection of data circuits within the physical service. This includes functions such as:
   - provide point-to-point and multipoint connections

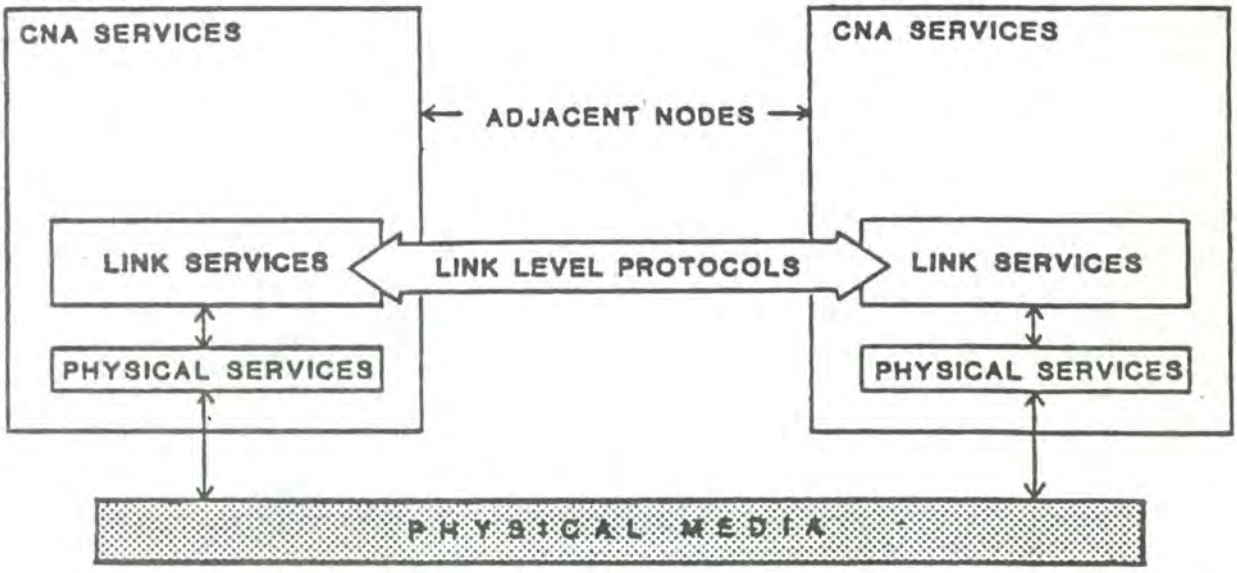


Figure 61 . Link Services Concept

6.4.7. Physical Services

CNA physical services provides services involving node-to-physical media control.

a) Provide the mechanical, electrical, functional and procedural characteristics to activate, maintain and deactivate physical connections for transmission between the users of this service
b) Provide, when possible, the desired quality of service between the users of this service
c) Provide control of the interconnection of data circuits to the users of this service

6.5. Protocol cuts

No information available.